

Article

Joint Eurocode-Compliance Classification and Reinforcement Regression with a Multi-Task Graph Neural Network Surrogate for Reinforced Concrete Predimensioning [†]

Nils Schäfer ^{1,*} , Uwe Rüppel ¹  and Joaquín Díaz ² 

¹ Institute of Numerical Methods and Informatics in Civil Engineering, Technical University of Darmstadt, 64287 Darmstadt, Germany; rueppel@iib.tu-darmstadt.de

² Department of Architecture and Civil Engineering, University of Applied Sciences Mittelhessen, 35390 Giessen, Germany; diaz@bau.thm.de

* Correspondence: schaefer@iib.tu-darmstadt.de

[†] This paper is an extended version of our paper published in *Advances in Information Technology in Civil and Building Engineering*, “Data-Driven Predimensioning: Applying Graph Neural Networks to Reinforced Concrete Design”, presented at ICCCBE 2024, Montreal, QC, Canada, 25–28 August 2024; pp. 181–190.

Abstract

Early-stage structural design requires rapid exploration of large design spaces, where the initial sizing of reinforced concrete members shapes downstream material use, cost, and the number of design iterations. Conventional predimensioning relies on experience and simplified formulae, while finite element analysis remains too slow for iterative use. This study presents a multi-task graph neural network surrogate that predicts per-element Eurocode compliance together with the required reinforcement for reinforced concrete slab-and-column buildings in one pass. A shared GraphSAGE encoder, trained on 2562 synthetic building graphs from automated finite element simulations, feeds one head for a compliance probability and another for reinforcement quantities. Because the rule-based Eurocode check is a hard pass-or-fail decision that does not vary smoothly with the design, the surrogate learns a continuous, differentiable compliance probability in its place, demonstrated for two representative criteria, one per element type, namely the $l/250$ deflection limit for slabs and the 4% reinforcement-ratio limit for columns. Across five random seeds, cost-sensitive focal-loss training that weights missed non-compliance above false alarms reached 90.9% balanced accuracy and held the share of non-compliant elements wrongly passed as compliant at 6.1% for columns and 1.6% for slabs, with a mean reinforcement error near 2% of the normalised target range. Inference averaged approximately 0.5 ms per building, between five and six orders of magnitude faster than the finite element analyses. A differentiable, multi-task graph surrogate therefore supports fast, cost-sensitive compliance screening for early-stage predimensioning, serving as a seed for gradient-based design exploration and a starting point for finite element verification.



Academic Editor: Adel Francis

Received: 30 May 2026

Revised: 22 June 2026

Accepted: 24 June 2026

Published: 29 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.

Keywords: graph neural networks; multi-task learning; surrogate model; reinforced concrete; predimensioning; early-stage design; Eurocode-compliance screening; cost-sensitive learning

1. Introduction

The construction sector accounts for a substantial share of global resource consumption and operational carbon emissions [1], with material-quantity decisions made in the earliest design stages shaping the environmental and economic footprint of a project throughout its life cycle. Predimensioning, the initial sizing of cross-sections and material quantities before

detailed structural verification, is the step at which these decisions are made. The quality of this step shapes subsequent material use, cost estimates, and the number of design iterations downstream. Traditional predimensioning relies on engineering experience, rules of thumb, and repeated manual recalculation. These workflows resist automation, which limits systematic exploration of the design space and consistent assessment of frequent design changes. Detailed finite element modelling can provide that feedback, but its computational cost is poorly suited to early-stage workflows where many alternatives must be screened within short time frames.

Machine learning offers a complementary route for this problem, particularly when the structural system can be represented as an interconnected set of load-bearing elements. Graph neural networks (GNNs) are well suited to this setting because they operate directly on graph-structured data and can encode load-transfer relationships between slabs, columns, and supports. Reviews of machine learning and surrogate models in structural engineering [2,3], graph neural networks for construction applications [4] and building operation and maintenance [5] document the rapid uptake of these methods in recent years. Existing studies have shown the value of these methods for structural layout generation [6], shear-wall design [7], and reinforcement optimisation [8]. Recent graph-based reinforced concrete contributions cover Eurocode-compliant beam-slab system sizing [9], heterogeneous frame sizing of beams, columns, and slabs as distinct node types [10], and physics-informed beam reinforcement detailing at element scope [11]. A detailed positioning of the present work relative to these and other graph- and machine-learning-based contributions follows in Section 2.

This article extends our earlier conference paper [12]. Compared with the conference version, which trained separate networks for compliance classification and reinforcement regression, this paper introduces a multi-task GraphSAGE architecture with a shared encoder for joint compliance and reinforcement prediction. It adds a cost-sensitive training regime that weights missed non-compliance higher than false alarms, paired with cost-sensitive reporting metrics that surface failure modes obscured by overall accuracy in imbalanced compliance classification. Additionally it introduces an operating-point selection framework that makes the conservatism-efficiency trade-off explicit as a designer decision.

Differentiability in this context refers to whether the per-element compliance signal exposes input gradients usable in gradient-based design refinement. Eurocode compliance, however, is evaluated through piecewise empirical rules whose outputs do not vary smoothly with structural inputs. Liao et al. [13] identify this non-differentiability of mechanical-performance and empirical-rule indicators as a key challenge for embedding code knowledge into the differentiable loss functions that drive end-to-end design optimisation. Prior gradient-coupled work for steel and reinforced concrete frames sidesteps this obstacle by differentiating a learned mechanical-response surrogate and enforcing the design limit on the regressed quantity, either through a penalty of fixed weight [14] or through Lagrangian relaxation, which adjusts the penalty weight during optimisation until the regressed quantity meets the limit [15]. The architecture proposed here instead learns a probabilistic per-element compliance signal in place of that rule-based check, so the signal responds smoothly to design changes and remains available as a gradient source. Gradient-free alternatives such as genetic algorithms or particle swarm optimisation remain feasible but evaluate a population of candidate designs at each iteration, which makes such population-based search computationally expensive [15]. A shared-encoder multi-task model that couples compliance classification and reinforcement regression returns input-feature gradients for both outputs from one backward pass, supporting joint refinement on either output or a weighted combination. The shared encoder is intended to couple the

two tasks through shared-representation regularisation following the multi-task learning rationale of Caruana [16].

Two complementary gaps motivate this work. The applied gap is practical, as the computational cost of finite element analysis makes iterative early-stage verification impractical, leaving designers reliant on conservative rules of thumb that both leave much of the design space unexplored and set only a sub-optimal starting point for the detailed design and analysis that follow, so reaching a material-efficient, compliant solution demands more costly iteration. The scientific gap is methodological, as current learned surrogates stop short of coupling per-element Eurocode-compliance classification with reinforcement regression in a single differentiable graph, the property that would turn fast screening into a gradient source for design refinement.

The proposed surrogate addresses both gaps, and its contribution has three aspects. First, a GNN architecture predicts Eurocode compliance and reinforcement quantities jointly for slabs and columns through one end-to-end differentiable model, implemented as a shared-encoder multi-task GraphSAGE. Second, a cost-sensitive reporting convention combines the pass-through-rate (PTR), the fraction of non-compliant elements that the surrogate misclassifies as compliant, with balanced accuracy under cost-sensitive focal-loss training, exposing missed non-compliance that overall accuracy obscures. Third, an operating-point selection framework makes the trade-off between conservative compliance screening and recall on the compliant class an explicit designer decision. This work establishes the differentiable per-element compliance signal as an architectural property and verifies it as a screening predictor, with its use within a gradient-based design loop deferred to future work (Section 5.4).

The research question is whether a single multi-task GNN can couple Eurocode-compliance classification with reinforcement regression in one differentiable model and deliver fast, cost-sensitive screening for early-stage reinforced concrete predimensioning. The surrogate is positioned as a screening tool that narrows the design space during early-stage exploration, with final structural verification remaining with established analytical and finite element tools. Compliance is operationalised through two representative Eurocode criteria, the $l/250$ deflection limit for slabs and the 4% reinforcement-ratio limit for columns, so the surrogate screens against these anchors while the complete Eurocode 2 design check remains part of the downstream verification stage. Figure 1 summarises this screening pipeline, from the reinforced concrete structure through its graph encoding to the per-element compliance verdict returned by the surrogate.

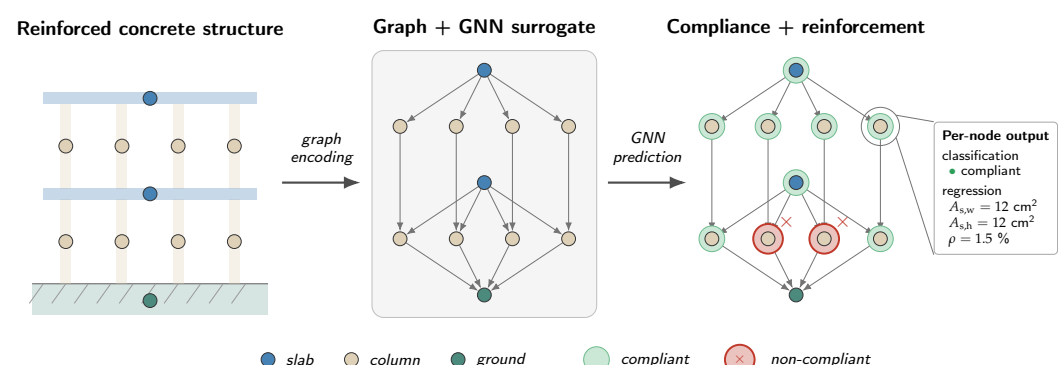


Figure 1. Overview of the proposed pipeline. A reinforced concrete structure is encoded as a graph, which the GNN surrogate maps to a per-element compliance verdict and reinforcement quantity prediction.

The remainder of this paper follows the six-step Design Science Research Methodology [17,18]. Problem identification and the definition of objectives are established in the

present section and substantiated by the gap statement in Section 2. Section 3 develops the surrogate artefact. Sections 4 and 5 demonstrate and evaluate it. Section 6 communicates the outcome.

2. Related Work

This section reviews the research areas that motivate the multi-task GraphSAGE surrogate proposed in Section 3. It introduces machine-learning surrogates for structural engineering (Section 2.1), the formulations through which reinforced-concrete sizing has incorporated code compliance (Section 2.2), and joint multi-task and differentiable formulations that aim to include code provisions into the design gradient (Section 2.3).

2.1. Graph- and Machine-Learning-Based Surrogates for Structural Engineering

Machine-learning surrogates have been applied across response prediction, reliability analysis, and structural health monitoring in structural engineering as an alternative to repeated finite element evaluation [2,3]. GNNs are particularly suited to skeletal structures, whose load paths follow the connectivity of structural members and their supports [14,19]. Three message-passing operators recur across this literature, namely graph convolution [20], graph attention [21], and GraphSAGE [22]. Parisi et al. [19] extend these operators through mechanics-informed edge-conditioned convolutions, and Li et al. [23] introduce heterogeneous-graph variants for the nonlinear analysis of mixed beam-and-plate structures. Chang and Cheng [14] established the bar-as-node graph representation and a millisecond-scale forward pass for steel beam-and-column frames under seismic loading, reporting a 6.8 ms surrogate evaluation against 13 s for the reference commercial analysis. The same graph-surrogate idea was later carried into truss surrogate modelling [24], truss optimisation [25], and tall-building response prediction [26]. Adjacent works address steel design through physics-informed neural networks [27] and early-stage steel sizing through machine-learning assistants [28], and automated code checking through graph-based learning for accessibility requirements [29]. A recent systematic review of machine learning for automated code checking [30] reports that graph neural networks have received limited attention and remain insufficiently investigated in this context, a gap the present surrogate addresses. At the building scale, Fei et al. [31] embed design-code minimum-reinforcement constraints into a GraphSAGE-based surrogate that regresses aggregate concrete and steel quantities for reinforced concrete shear-wall buildings, though it returns a single building-level estimate without per-element compliance verdicts. Relational context has also been approximated through hand-engineered neighbour descriptors fed to dense networks [32], whereas a graph formulation lets the encoder learn which member relationships to aggregate directly from the topology. None of these surrogates emits a per-element code-compliance signal coupled with a reinforcement quantity at the building scale, which motivates the reinforced-concrete-specific literature reviewed next.

2.2. Element- and Building-Level Reinforced Concrete Sizing Under Code Compliance

Reinforced concrete sizing must satisfy code requirements while remaining tractable for iterative design exploration. Recent neural-network models report higher accuracy than closed-form empirical expressions for column capacity [33], reinforcing the case for data-driven surrogates in reinforced concrete design. Class-imbalance and cost-sensitive learning likewise remain active across recent structural and building-engineering work, from class-weight-optimised classification of reinforced concrete failure modes [34] to imbalance-aware deep learning for structural monitoring [35]. We organise prior work into three families distinguished by how code compliance enters the learning pipeline.

A first family lets the surrogate predict a structural quantity and delegates compliance verification to an outer search loop. Li et al. [8] combine a GNN with an exploratory genetic algorithm for clash-free rebar detailing, reporting a 75% to 90% reduction in computation time over conventional optimisation. Shear-wall sizing and layout are addressed through surrogate-assisted metaheuristic search in Fei et al. [36] and Lou et al. [37]. Similarly, Zhang et al. [38] pair a neural-network capacity surrogate for composite columns with an outer multi-objective genetic search that queries the surrogate to evaluate candidate designs rapidly.

A second family couples compliance into training itself. Jayaweera et al. [9] embed code-aware sizing targets under EN 1992-1-1, Verduzco et al. [11] add constructability-aware physics-informed losses under the Hong Kong Code of Practice, and Qin et al. [10] train a heterogeneous GNN on Chinese-code-designed design-institute drawings, achieving sub-second sizing to within roughly one 50 mm size module of engineer designs.

A third family regresses compliance itself as a per-structure resistance-to-demand ratio, illustrated by Kuhn et al. [39] for the pre-assessment of existing reinforced-concrete frame bridges and supporting maintenance prioritisation on the most demand-critical structures. Across these families, the surrogate itself does not emit a per-element compliance probability that varies smoothly with design inputs, so a learned classification head coupled to a reinforcement regression head in a single differentiable graph remains unaddressed in the literature known to the authors.

2.3. Joint Multi-Task and Differentiable Formulations

Multi-task learning provides one route to joint formulations through shared representations that regularise related prediction tasks toward features useful for both [16].

A smaller body of work targets gradient-coupled and differentiable formulations that integrate learned predictions with downstream optimisation. Zhang et al. [15] report a differentiable optimisation framework for reinforced concrete frames, combining Gumbel-Softmax discrete-section parameterisation with Lagrangian-relaxation gradient methods (including dual gradient ascent) on a concrete-cost objective under drift and lateral-stiffness regression constraints. Chang and Cheng [14] introduced an earlier differentiable formulation for steel frames, in which a learned drift-response surrogate drives a gradient-based section selector through a storey-level drift-limit penalty applied to the regressed drift response. An auxiliary drift-exceedance classifier supports simulator training in their pipeline but does not feed the design gradient. These gradient-coupled formulations show that the design loop itself can become differentiable, while the code-of-practice check that decides design admissibility remains outside the gradient path. The survey of Liao et al. [13] reviews data representations, generative algorithms, and evaluation methods across multiple building-structure types such as shear walls and frames, and identifies the requirement for differentiable constraint-based loss functions as a recurring bottleneck for end-to-end optimisation.

Across the reviewed graph-based surrogates, compliance-aware sizing formulations, and gradient-coupled approaches, three properties are, to the best of the authors' knowledge, not jointly addressed. First (i), no surrogate predicts per-element compliance across both slab and column elements within one building graph. Second (ii), no work pairs Eurocode-compliance classification with reinforcement regression in a single shared-encoder model. Third (iii), existing gradient-coupled formulations treat code constraints as regression-and-threshold post-checks, while a learned per-element compliance classification that supplies a differentiable pass-or-fail signal remains absent. The present work addresses these three points within a single multi-task GraphSAGE surrogate. Section 3.1 and Section 3.2 establish point (i) by constructing per-building graphs over slabs and

columns from finite element analyses of 2562 layouts. Section 3.3 addresses points (ii) and (iii) through a shared graph encoder paired with a compliance-classification head and a reinforcement-regression head, with both heads coupled in one differentiable graph and discussed further in Section 5. The surrogate further delivers sub-millisecond inference suitable for iterative early-stage screening, reported in Section 4.3.

3. Materials and Methods

This section presents the design and development of the surrogate model artefact within the methodology introduced in Section 1. The workflow comprised four stages, namely synthetic data generation, automated finite element method (FEM) simulation, graph construction, and supervised multi-task learning for joint compliance classification and reinforcement regression. Figure 2 presents these four stages together with the concluding held-out evaluation as a single research-process flow, situating the methods within the Design Science Research process introduced in Section 1. Section 3.1 describes the data generation and finite element simulation, Section 3.2 the graph representation and training setup, and Section 3.3 the multi-task GraphSAGE architecture and evaluation metrics.

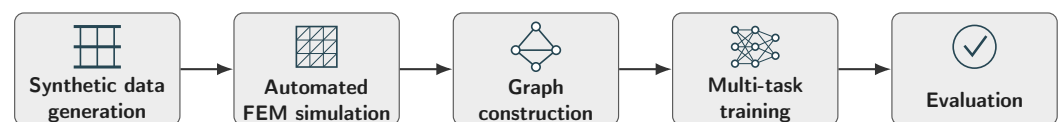


Figure 2. Research process of the study, from synthetic data generation through automated finite element labelling, graph construction, and multi-task training to evaluation.

3.1. Data Generation

The starting point was the 4,000 building structures introduced by Chang and Cheng [14]. These source structures were adapted from steel to reinforced concrete and converted to metric dimensions for Eurocode-oriented slab and column design. The adapted layouts used storey heights of 3.5 m, slab spans of 5.0 to 7.5 m, concrete strength class C30/37 XC1, slab thicknesses of 18 to 34 cm, and square column cross-sections of 30 to 60 cm. A base storey was replicated up to ten storeys. The resulting structural systems contained reinforced concrete slabs and columns only.

Roof loads were defined as $G = 2 \text{ kN/m}^2$ and $Q = 1 \text{ kN/m}^2$. Floor loads were defined as $G = 1.2 \text{ kN/m}^2$ and $Q = 5 \text{ kN/m}^2$. The stated dead loads are superimposed dead loads applied in addition to the element self-weight, which the finite element solver generates automatically from member geometry and material density. Load combinations were generated for the ultimate and serviceability limit states, with partial factors of 1.35 on permanent actions and 1.5 on variable actions for the ultimate combination and a quasi-permanent factor of 0.3 on variable actions for the serviceability combination. Slab checks followed the quasi-permanent serviceability combination with the $l/250$ deflection criterion, while column design considered stability, crack width limitation, and load transfer. The source layouts originate from steel frames whose floor regions may include openings, whereas the finite element pipeline can define only closed slab fields without openings. Layouts whose floor regions could not be closed into consistent slab boundaries were removed in a geometric pre-screening before simulation, and a second pass after simulation removed cases in which the finite element analysis or the subsequent reinforcement calculation did not converge. The effective analysed dataset comprised 2562 building graphs from the original 4000 source structures. Both passes act on source geometry and on solver convergence and are applied before any model training, so the excluded cases are characterised entirely by their source geometry and the feasibility of the simulation. The principal systematic effect

of the reduction is a retained dataset of closed slab fields without openings, a coverage limit whose implications for real building stock are discussed in Section 5.3.

Columns were created from the endpoints of the scaled vertical lines in each layout. Ground-floor columns were fixed, and random initial cross-sections were assigned to reduce bias in the generated data. Buckling-length coefficients of 0.7 and 1.0 were used according to the support conditions. Slabs were created from ordered horizontal boundary lines and assigned both slab thicknesses and deflection profiles. The deflection profile was derived from the shortest support distance (l) and the $l/250$ serviceability requirement. Identifiers for buildings and structural elements were retained throughout the workflow to preserve traceability between geometry, simulation outputs, and graph data.

FEM-Simulation

The simulation configuration followed Autodesk Robot Structural Analysis Professional 2023 (RSA) conventions for slab discretisation, support-singularity handling, and Eurocode-based serviceability checks, with response quantities serving as labels for GNN training. The automation was implemented through the RSA application programming interface in IronPython 2.7. A predefined RSA template loaded consistent project preferences, reinforcement settings, and Eurocode-based design assumptions. The finite element mesh was standardised to 30 cm. This size kept the discretisation below twice the minimum slab thickness, above half the maximum slab thickness, and ensured at least eight nodes between adjacent supports. The simulation sequence comprised three steps. The first step initialised the structural model and generated the finite element representation. The second step computed slab reinforcement and deflection, using the analytical calculation option and singularity reduction at point supports. The third step computed column reinforcement under stability and crack-width constraints, including a characteristic crack-width limit of $w_k \leq 0.4$ mm for exposure class XC1 under the quasi-permanent action combination (EN 1992-1-1, Table 7.1N) [40]. Exported slab results included reinforcement in the upper and lower layers in both directions and the resulting deflection. Exported column results included reinforcement along width and height, reinforcement ratio, and the associated design forces and moments. The pipeline also stored error flags, computation times, geometric identifiers, shortest support distances, assigned deflection profiles, and building-level counts of slabs and columns.

Preparation and Analysis

Post-processing transformed the simulation outputs into slab-level, column-level, and building-level CSV files for graph construction and model training. These exports preserved the correspondence between finite elements, structural elements, and building instances across all preprocessing steps. Binary classification labels for slabs were derived from compliance with the $l/250$ deflection criterion (EN 1992-1-1, § 7.4.1 (4)) [40]. Binary classification labels for columns were derived from the 4% maximum-reinforcement-ratio limit (EN 1992-1-1, § 9.5.2 (3)) [40]. The full Eurocode 2 design check is not reproduced in the surrogate. The $l/250$ deflection limit for slabs and the 4% reinforcement-ratio limit for columns serve as single representative compliance anchors for simplified slab-and-column predimensioning under Eurocode. The deflection limit captures a serviceability mode relevant for the slab configurations considered here, while the reinforcement-ratio limit captures a detailing bound on column cross-section sizing. For the point-supported slabs modelled in this study, punching shear at the slab-column connection (EN 1992-1-1, § 6.4) [40] is a further limit state that can govern slab thickness, and its omission from the compliance label is a deliberate scope restriction examined in Section 5.3. The same two compliance anchors were used in the conference precursor of this work [12]. Regression

targets for slabs were defined as the mean reinforcement values of all finite elements within each slab. Regression targets for columns remained element-specific. The final slab targets were lower-layer X, lower-layer Y, upper-layer X, and upper-layer Y reinforcement in cm^2/m . The final column targets were reinforcement along width, reinforcement along height, and reinforcement ratio. All regression targets were min-max normalised to the unit interval before training, so the reported mean reinforcement errors are expressed on that normalised scale.

3.2. Graph Representation and Training

Graph Creation

Simulation outputs were converted into graph data with Python 3.12 and NetworkX 3.6.1. Each building was represented as a directed graph whose nodes corresponded to slabs and columns. Each node carried a 32-dimensional feature vector encoding coordinates, cross-sections, structure type, load information, storey assignment, error states, deflection criteria, and reinforcement-related quantities. Cross-sections were represented through one-hot encoding, while continuous quantities were stored as numerical features. A pseudo-ground node with zero-valued features was added to anchor the load path at foundation level. Directed edges connected slabs to columns, columns to columns, and columns to the pseudo-ground node. This directionality follows the principal gravity load path of the structure. It therefore supplies physically meaningful connectivity for message passing and aligns the graph representation with structural load transfer. The graphs were stored as lists of dictionaries so that geometry, connectivity, storey membership, and target values remained accessible during preprocessing and training.

The graph encoding is illustrated in Figure 3, which maps a representative reinforced concrete building to its directed gravity-load graph. The sensitivity of the model to this directional choice is reported in Section 4.4, based on a five-seed re-training of the same architecture on an undirected variant of the graph data.

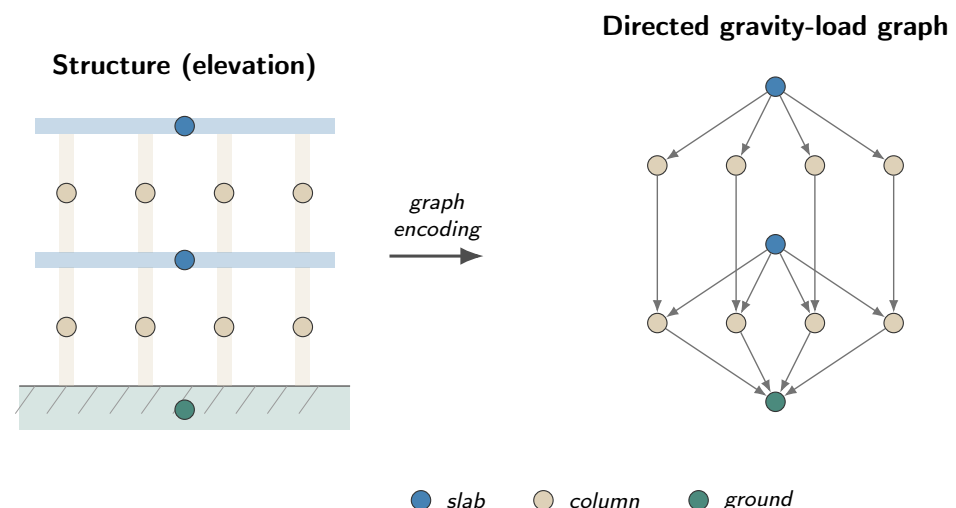


Figure 3. Mapping of a representative two-storey structure (left) to its directed graph (right). Nodes represent floor slabs, columns, and a single pseudo-ground. Directed edges follow the principal gravity load path from slabs through columns to the ground.

Training Setup

Training was conducted in Google Colab on NVIDIA T4 GPUs with PyTorch 2.11 and PyTorch Geometric 2.7, with scikit-learn 1.8 used for the classification and regression metrics. Preprocessing assembled node-specific feature vectors, generated directed edge tensors, and converted each building graph into a PyTorch Geometric ‘Data’ object. Con-

tinuous features such as coordinates were min-max scaled to limit the dominance of large raw values during optimisation. The seven regression targets were merged into one output vector and scaled per index. The same graph instance also carried the binary compliance labels used for classification. Table 1 summarises the per-node prediction targets for all surrogate architectures.

Table 1. Per-node prediction targets for all surrogate architectures. Every node receives one binary compliance score and a seven-dimensional regression vector, with slabs and columns each assigned their own reinforcement targets and both classification thresholds taken from Section 3.1.

Label	Per-Node Dim	Quantity
Classification	1 (binary)	Compliance score
Regression (slab)	4 of 7	Reinforcement in lower-X, lower-Y, upper-X, and upper-Y layers (cm^2/m)
Regression (column)	3 of 7	Reinforcement along width (cm^2) Reinforcement along height (cm^2) Reinforcement ratio (%)

Reference Baselines

The operator comparison reported in Section 4 evaluates three message-passing operators as single-task classifiers, namely a Graph Attention Network (GAT) [21], a Graph Convolutional Network (GCN) [20], and the GraphSAGE operator [22], with the GAT and GCN variants introduced in Schäfer et al. [12]. The three message-passing operators differ in how each node aggregates its neighbourhood. GCN aggregates each node and its neighbours together through a symmetrically normalised mean of their feature vectors, GAT learns attention weights over the neighbours and combines them through a weighted sum, and GraphSAGE separates the node's own features from the neighbour aggregate through a dedicated self-projection that preserves node-level structural attributes such as cross-section, span, and load. All three operator baselines use three message-passing layers with hidden dimensions 128 and 64 and are trained under the cost-sensitive focal-loss regime described in Section 3.3, so the comparison isolates the operator at matched depth, task structure, and loss.

Separate single-task GraphSAGE classification and regression variants, denoted SAGE_STL and trained under the same protocol, isolate the contribution of the joint multi-task formulation at the fixed GraphSAGE operator and depth.

3.3. Multi-Task GraphSAGE Architecture

The joint multi-task learning variant of GraphSAGE proposed here, denoted SAGE_MTL in the remainder of the paper, trains a compliance classification head jointly with a reinforcement regression head on a shared encoder under a cost-sensitive focal-loss regime. Compliance and reinforcement quantities are decoded from the same shared latent representation, coupling both outputs through the encoder weights. For columns, the binary compliance label is the threshold function of a regressed quantity, the 4% reinforcement-ratio limit in EN 1992-1-1, § 9.5.2 (3). For slabs, the reinforcement target and the deflection that triggers the $l/250$ classification both arise from the same finite element solution. The proposed architecture comprises a two-layer shared GraphSAGE encoder [22] with hidden dimensions 128 and 64, followed by two parallel GraphSAGE convolutional heads. The shared encoder processes the input graph into 64-dimensional node representations, which are then passed in parallel to the two heads, so each output path traverses three message-passing layers and matches the depth of the GAT and GCN operator baselines. The classification head outputs a single compliance score per node for binary compliance prediction, while the regression head outputs seven values per node for reinforcement quan-

ties. The full network contains approximately 25,800 parameters. Figure 4 summarises the resulting architecture.

Slab nodes are assigned the slab-reinforcement targets and the slab compliance label, column nodes are assigned the column-reinforcement targets and the column compliance label, and the pseudo-ground node receives a sentinel classification label of -1 that is masked out of the loss. Within the seven regression dimensions, irrelevant element-type targets are zero-filled, and the L_1 loss is computed across the full output vector at every non-pseudo-ground node. The mean absolute error reported in Section 4 therefore averages the absolute reinforcement deviation across all seven outputs at every evaluated node for both SAGE_MTL and the reference baselines of Schäfer et al. [12], which keeps these values directly comparable.

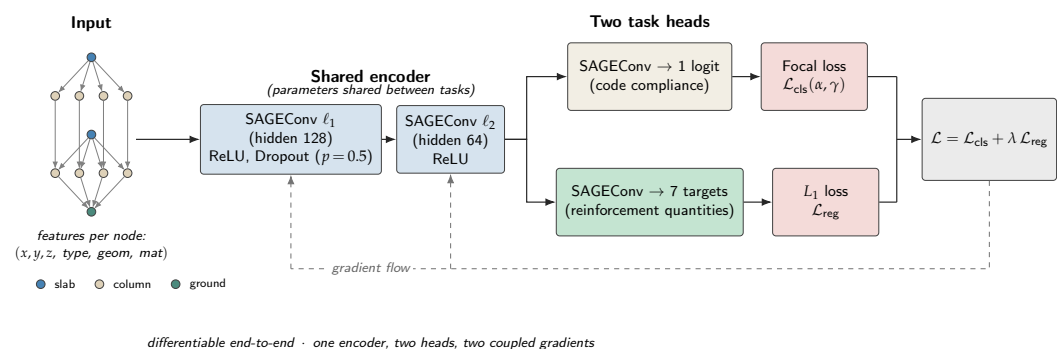


Figure 4. SAGE_MTL architecture. Two shared GraphSAGE layers encode the directed gravity-load graph into 64-dimensional node embeddings, which a classification head and a regression head decode in parallel into a per-node Eurocode-compliance score and seven reinforcement targets. Gradients from both heads flow back through the shared encoder.

Class-Imbalance and Cost-Sensitive Learning

Class-imbalanced safety-critical screening benefits from cost-sensitive learning, which encodes asymmetric error costs directly into the training objective [41,42]. The cost asymmetry here is structural. A non-compliant element passed as compliant propagates into downstream design decisions and undermines the screening function, whereas a compliant element flagged as non-compliant only triggers an unnecessary finite element verification. Focal loss [43] realises this asymmetry through class-balancing weight α , which scales the minority (non-compliant) class, while a focusing parameter γ concentrates gradient updates on difficult examples.

Training optimises a joint loss of the form $L = L_{\text{cls}} + \lambda \cdot L_{\text{reg}}$, where the classification term L_{cls} is a binary focal loss [43] with focusing parameter $\gamma = 1.0$ and class-balancing parameter α set automatically to the empirical ratio of majority (compliant) to minority (non-compliant) samples in the training set, yielding $\alpha \approx 21.4$. The regression term L_{reg} is the mean absolute error over the seven reinforcement targets, with the regression-loss weight λ fixed at 1.0 for all reported runs. Optimisation uses the Adam optimiser [44] with decoupled weight decay, learning rate 5×10^{-4} , weight decay 1×10^{-4} , and gradient clipping at 1.0, on mini-batches of 16 building graphs for a maximum of 30 epochs, with early stopping on the validation-set column pass-through-rate after a patience of 10 epochs. The dataset is partitioned into training, validation, and test sets in an 80/10/10 ratio with a fixed random state of 22, identical across all five reported seeds.

Classification outputs are converted into binary compliance decisions through a fixed threshold of 0.50, without validation-set tuning, to avoid threshold optimisation on the test set. The conservatism-efficiency trade-off at the operating point is examined in Section 4.3.

Evaluation Metrics

Per-element classification outputs are summarised through the four confusion-matrix counts TP, FN, FP, and TN, where the positive class is the compliant class. A true positive is a compliant element correctly screened as compliant and a true negative a non-compliant element correctly identified, whereas a false positive is a non-compliant element incorrectly passed through the screening filter and a false negative a compliant element flagged as non-compliant. Because the positive class is the compliant class, the safety-critical error of a missed non-compliant element is a false positive that propagates into downstream design decisions, whereas a false negative only triggers an unnecessary finite element verification. The aggregate classification metrics reported in Section 4 follow from these counts,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{BalAcc} = \frac{1}{2} \left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}} \right) \quad (4)$$

while the safety-critical screening error is reported per element class through the specificity and pass-through-rate,

$$\text{Specificity}_c = \frac{\text{TN}_c}{\text{TN}_c + \text{FP}_c} \quad (5)$$

$$\text{PTR}_c = \frac{\text{FP}_c}{\text{FP}_c + \text{TN}_c} = 1 - \text{Specificity}_c \quad (6)$$

where the subscript c ranges over the slab and column element classes. Overall accuracy reports the share of correctly classified elements, precision on the compliant class the share of true compliant elements among those predicted compliant, and recall on the compliant class, equivalently sensitivity, the share of compliant elements correctly recovered. The F1 score is omitted because it is the harmonic mean of precision and recall on the positive class, which here is the majority compliant class, so its value is governed by the abundant compliant elements and conceals the asymmetric false-positive cost that the pass-through-rate isolates. Balanced accuracy, the unweighted mean of recall and specificity, corrects overall accuracy for the class imbalance characterised in Section 4.1, while the pass-through-rate PTR_c gives the share of non-compliant elements of class c predicted as compliant and corresponds to the safety-critical screening error. Columns form the extreme minority class at 2.5% non-compliant nodes, so the column pass-through-rate is the binding screening metric, with a missed non-compliant column both the rarest and the most safety-critical outcome. Regression performance is reported as the mean absolute error over the seven reinforcement targets defined in Table 1, averaged across all evaluated nodes. The receiver operating characteristic (ROC) area-under-curve summarises the class separation achieved by the predicted compliance probabilities independently of the decision threshold.

Architecture and Loss Selection

The operator and loss formulation were selected from an exploratory factorial across four convolutional operators (GCN [20], GAT, two attention-based variants, and GraphSAGE) and four imbalance-aware losses (binary focal loss together with three reweighting variants), trained under the same data split and training protocol. GraphSAGE was retained from this factorial for its explicit self-term, in which a node's own features pass through a

separate linear projection from the neighbourhood aggregate, preserving the node-level structural attributes such as cross-section, span, and load.

4. Results

The reported results refer to the held-out test set. They are organised around the analysed dataset and its class imbalance, the classification and regression performance of SAGE_MTL against reference baselines, the class separation and inference speed that complement these metrics, and a set of behavioural and sensitivity analyses on the trained model.

4.1. Dataset Characterisation and Class Imbalance

The reinforced concrete dataset is strongly imbalanced on the compliance axis, and this imbalance shapes how screening performance must be reported. Figure 5 reports the share of compliant and non-compliant nodes by element type, across the analysed dataset and across the three data partitions. Across the full dataset, 29.1% of slab nodes are non-compliant against 2.5% of column nodes, an order-of-magnitude gap that is preserved across training, validation, and test partitions. Slabs make up only 7.4% of the node count, while their within-type non-compliance rate exceeds that of columns by an order of magnitude.

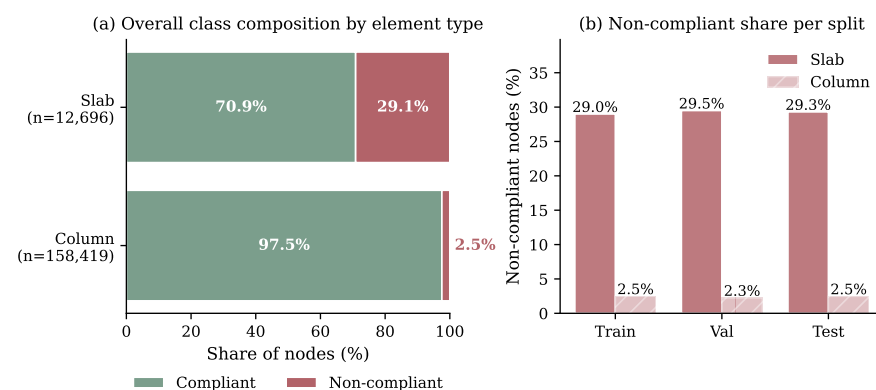


Figure 5. Composition of compliant and non-compliant nodes by element type, across the full dataset (panel (a), with absolute node counts) and across the three data partitions (panel (b), share of non-compliant nodes).

4.2. Compliance Classification and Reinforcement Prediction

Given the class imbalance characterised in Section 4.1, a classifier can reach high overall accuracy while failing to identify non-compliant columns. The GAT baseline of Schäfer et al. [12] under plain binary cross-entropy reaches 96.99% overall accuracy on the same test partition, yet attains a column specificity of 0.00%, so every non-compliant column passes the screening filter undetected. A loss-weighting variant using $\text{pos_weight} \in \{0.05, \dots, 0.09\}$ across five seeds recovers column specificity transiently during training, yet no operating point combines this recovery with the overall accuracy of the plain BCE baseline.

Under the cost-sensitive focal-loss regime described in Section 3.3 (focusing parameter $\gamma = 1.0$, class-balancing parameter $\alpha \approx 21.4$), the three convolutional operators GAT, GCN, and GraphSAGE were trained as single-task classifiers across five random seeds $\{22, 42, 123, 2024, 7\}$ on the same data split and at matched depth. Table 2 reports this operator comparison alongside the Eurocode-based finite element ground truth obtained from Autodesk Robot Structural Analysis. Single-task GAT attains a column pass-through-rate of 0.176 and GCN 0.307, against 0.060 for single-task GraphSAGE, which reaches the lowest

column pass-through-rate and the highest balanced accuracy of the three operators and motivates the operator selection of Section 3.3.

Table 2. Operator comparison under matched single-task classification training, alongside the Eurocode-based finite element ground truth. Surrogate values are mean and standard deviation across five random seeds on the held-out test set. PTR is the pass-through-rate, the share of non-compliant elements predicted as compliant, and GAT, GCN, and FEM denote the Graph Attention Network, Graph Convolutional Network, and finite element method.

Metric	FEM Ground Truth	GAT	GCN	GraphSAGE
Overall accuracy (%)	100.00	71.09 ± 6.36	73.80 ± 2.90	85.47 ± 0.53
Balanced accuracy (%)	100.00	80.70 ± 2.54	78.15 ± 2.51	90.66 ± 0.22
Precision compliant (%)	100.00	99.48 ± 0.10	99.04 ± 0.30	99.82 ± 0.01
Column PTR	0.000	0.176 ± 0.044	0.307 ± 0.076	0.060 ± 0.004
Slab PTR	0.000	0.004 ± 0.004	0.042 ± 0.048	0.015 ± 0.001

Building on the GraphSAGE operator, the multi-task SAGE_MTL reaches a balanced accuracy of 90.89% and a column pass-through-rate of 0.061 across the five seeds, with the slab pass-through-rate held at 0.016 and a mean reinforcement error of 10.77×10^{-3} on the seven reinforcement targets, as reported in Table 3. Its overall accuracy of 86.0% falls below the balanced accuracy by design, because catching the few non-compliant elements means flagging some of the many compliant ones and these extra false alarms lower the overall share of correct predictions while leaving the balanced accuracy high.

Table 3. Multi-task ablation and performance at a fixed GraphSAGE operator and depth. The single-task variants are trained under the SAGE_MTL protocol, with values reported as mean and standard deviation across five random seeds on the held-out test set.

Model	Balanced Accuracy (%)	Column PTR	Slab PTR	Mean L_1 ($\times 10^{-3}$)
SAGE_STL (classification)	90.66 ± 0.22	0.060 ± 0.004	0.015 ± 0.001	n/a
SAGE_STL (regression)	n/a	n/a	n/a	9.41 ± 0.55
SAGE_MTL (joint)	90.89 ± 0.10	0.061 ± 0.001	0.016 ± 0.000	10.77 ± 1.19

Table 3 isolates the multi-task contribution by comparing SAGE_MTL against single-task GraphSAGE variants trained under the same protocol, operator, and depth. The single-task classification model matches the joint model on the column pass-through-rate, 0.060 against 0.061, while the single-task regression model reaches a mean reinforcement error of 9.41 against 10.77×10^{-3} , a difference comparable to the variation between the five training runs. The joint model matches the two single-task specialists on both the compliance classification and the reinforcement regression, while producing both outputs from one shared encoder.

Restricted to the dimensions relevant to each element type, which removes the zero-filled targets, the per-building reinforcement error averages about 2% of the normalised target range and reaches a maximum near 8% across the test buildings.

The columns the surrogate misses are consistently borderline cases whose reinforcement ratio only marginally exceeds the 4% limit, so the more severe violations are reliably caught and no gross over-reinforcement passes the screen. Because they are predicted compliant only by a narrow margin and recur as the same small group across all five training runs, the residual screening error is a predictable boundary effect that a higher compliance threshold reduces by flagging most of these columns for verification without retraining.

4.3. Class Separation and Inference Speed

The operating point, namely the probability threshold that converts the per-element compliance prediction into a binary class label, was fixed at 0.50 throughout the preceding evaluation. Operating-point-independent class separation complements the threshold-dependent metrics of Tables 2 and 3. The ROC area-under-curve reaches 0.967 on the overall test set and is essentially constant across the five seeds, so the predicted scores rank compliant above non-compliant elements largely independently of the threshold. Figure 6 shows the corresponding curve with the fixed operating point of 0.50 marked.

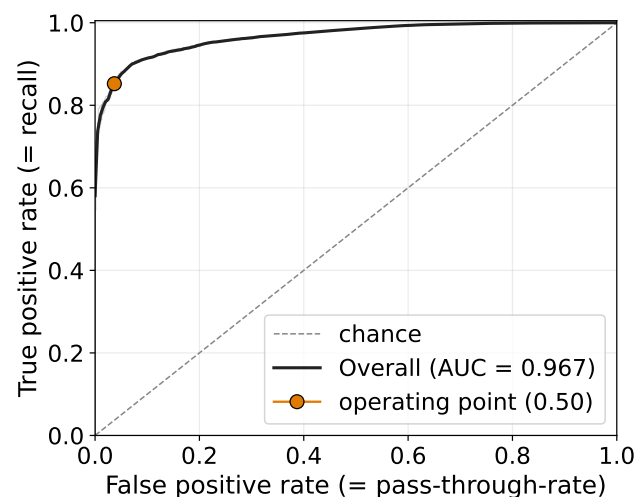


Figure 6. Receiver operating characteristic of SAGE_MTL on the held-out test set across the five random seeds (thin lines per seed, bold line the mean). The marker is the fixed operating point at threshold 0.50, where the false-positive rate equals the pooled slab-and-column pass-through-rate.

Figure 7 resolves the same operating point into the predicted-score distributions of the two true classes. The narrow overlap region in Figure 7 is the band in which an operating-point shift trades pass-through-rate against compliant recall without retraining.

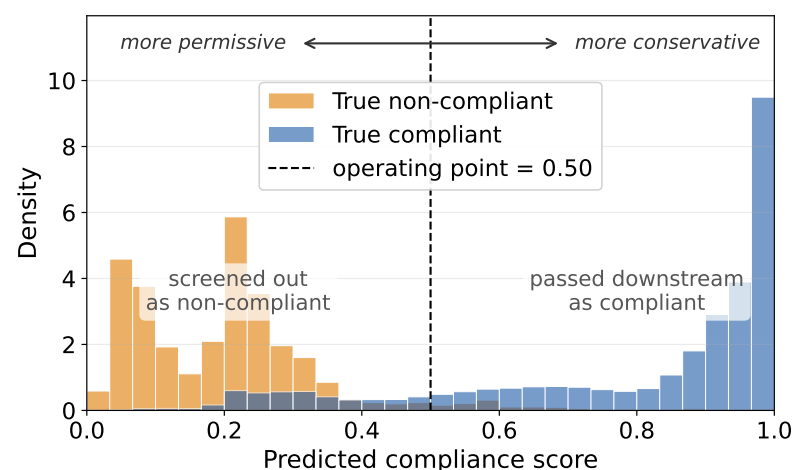


Figure 7. Per-class density of the predicted compliance score on the held-out test set, pooled across the five random seeds. The dashed line marks the default operating point at 0.50, beyond which scores are passed downstream as compliant.

SAGE_MTL produces the joint classification and regression output in 0.48 ms per building on average and up to 0.79 ms for the largest building of 433 nodes, timed as the median of 50 forward passes per building at batch size 1 on an AMD Ryzen 7 PRO 7840U CPU. FEM simulations on the same dataset required about 2 min on average and up to

25 min for the largest buildings. The surrogate latency is therefore between five and six orders of magnitude below the finite element runtime.

4.4. Model Behaviour and Sensitivity Analyses

Three further analyses probe the trained model beyond the reported performance metrics, namely the gradient signal available for design refinement, the agreement between the learned classification head and a regression-threshold substitute for columns, and the sensitivity of the screening profile to edge directionality.

Gradient ascent on the continuous input features of the trained model probed the differentiability of the compliance signal. Across the 292 non-compliant columns sampled near the decision boundary, a single ascent step raised the predicted compliance probability for every column, by a mean of 0.068.

Because the column compliance label is by construction a fixed threshold on the regressed reinforcement ratio, a separate learned classification head is justified only if it improves on a direct threshold of that regressed quantity. A fixed 4% threshold on that ratio therefore served as a substitute for the learned classification head on the same column nodes. The substitute raised the column pass-through-rate to 1.00, against 0.061 for the learned head, while the two routes agreed on 86% of test-set column nodes. On the remaining nodes the classification head flagged non-compliant columns that the regression threshold missed in every case, with no instance of the reverse.

Re-training the same architecture across five seeds on an undirected variant of the graph, with all other settings held fixed, assessed the sensitivity to the directed gravity-load encoding. The undirected encoding raised the column pass-through-rate by 2.6 percentage points, from 0.061 to 0.087, while leaving the mean regression L_1 unchanged.

5. Discussion

The multi-task surrogate predicts Eurocode compliance and reinforcement quantities jointly through one end-to-end differentiable forward pass, and the balanced accuracy of 90.9% reported in Section 4 indicates that the architecture provides cost-sensitive compliance screening for early-stage reinforced concrete predimensioning. The remainder of this section evaluates that result against the screening objectives of Section 1, organised along four axes. Section 5.1 examines the differentiable surrogate role in gradient-coupled design exploration. Section 5.2 treats operating-point selection together with element-type prediction behaviour. Section 5.3 relates the surrogate scope to final verification and to the bounds of the underlying data. Section 5.4 outlines the future research directions opened by these properties.

Table 4 positions the present surrogate against the closest prior graph-neural-network surrogates along the axes examined in the following subsections, namely the differentiable per-element compliance signal, the joint compliance and reinforcement outputs, the cost-sensitive screening behaviour, and the scope and validation basis.

A direct numerical comparison across these works is precluded by their differing structural systems, datasets, and reported metrics, so the table positions the works along qualitative capability axes. Across the compared surrogates, code compliance enters the learning pipeline only in indirect forms, through an outer search loop, a regression target paired with an external threshold check, a physics-informed training loss, or an embedded minimum-reinforcement rule, none of which exposes a per-element compliance decision predicted by the model itself. A work is rated partial on the differentiable axis when its pipeline is gradient-coupled and differentiable yet stops short of a learned per-element compliance signal, as in the drift-response loop of Chang and Cheng [14] and the physics-informed loss of Verduzco et al. [11], and absent when admissibility is decided only

outside any gradient path. The learned per-element classification that distinguishes the present approach was introduced in the authors' earlier conference surrogate [12], and the contribution here is the cost-sensitive training that weights missed non-compliance above false alarms, which the comparison shows to be absent from prior reinforced concrete sizing work. This emphasis follows from the predimensioning setting, where advancing a non-compliant member into later design stages carries a heavier consequence than an unnecessary verification, so the direction of the screening error matters more than overall accuracy. The present evidence rests on a synthetic dataset, whereas the comparison includes work that draws on real-project data [10,31], which keeps independent real-world validation a necessary step before deployment.

5.1. Differentiable Surrogate as a Foundation for Gradient-Based Predimensioning

Liao et al. [13] identify the non-differentiability of structural-design evaluation as an open research problem for embedding code knowledge into end-to-end design optimisation. Gradient-coupled surrogates have been demonstrated for steel seismic response [14] and for reinforced concrete frames [15], but neither work emits a per-element compliance probability. The differentiable pipeline therefore ends before the code check that governs the screening decision. The architecture proposed here addresses that gap by training a compliance classification head and a reinforcement regression head jointly on a shared encoder, so that a learned per-element compliance probability and learned reinforcement quantities are produced together through one end-to-end differentiable forward pass. Across the ablation in Table 3, the joint model matches the two single-task specialists on both tasks, consistent with the multi-task learning principle that related tasks share a useful representation [16,45], so a single shared encoder supplies both the compliance and the reinforcement outputs.

The shared forward pass supplies the input gradients on both outputs that a closed gradient-based design loop would require. Continuous per-element design variables enter through the regression head outputs and through the input node features, and gradients on both outputs propagate through the shared encoder to an outer constrained gradient-based optimiser of the kind used in related work [15]. The compliance gradient remains informative under the saturated sigmoid, as the gradient-ascent probe of Section 4.4 shows for the columns sampled near the decision boundary. The reinforcement head is linear in the shared embedding, so its input gradient does not saturate and contributes to the combined backward pass at every node. Deep in the saturated regime the probability moves little per step, the expected behaviour of a focal-loss-trained classifier that a closed-loop deployment would address through an added calibration layer, while near the decision boundary where a design loop operates the gradient supplies a usable update direction. Operationalisation of the full loop is left to future work.

The differentiable architecture extends naturally toward broader Eurocode coverage. Each additional code provision adds an output on the same shared encoder, realised either as a further head or as an extra unit of a multi-label compliance head, so the multi-task formulation couples related provisions through a shared representation. The reported criteria stand as two representative compliance anchors for the simplified predimensioning context considered here, while a complete treatment of a flat slab-and-column structure would add further governing provisions such as punching shear. A complementary route is offered by Zhang et al. [15], who relax the code constraints through differentiable approximations of the non-smooth constraints under a Lagrangian formulation, an approach compatible with the multi-head formulation here.

The same sub-millisecond inference speed makes it practical to screen orders of magnitude more design alternatives within the time budget of early-stage predimensioning

than finite element workflows allow, a throughput advantage aligned with the material-efficiency and embodied-carbon motivations set out in Section 1.

Table 4. Positioning of the proposed surrogate against the closest graph-neural-network surrogates for structural sizing and compliance. Diff. denotes a differentiable per-element compliance signal and Cost-s. cost-sensitive training, with ✓ present, ~ partial, and × absent.

Work	Scope	Outputs	Compliance Handling	Diff.	Cost-s.	Validation
Chang & Cheng [14], GNN	Steel frames	Storey drift, section sizing	Drift-limit penalty in a differentiable design loop, story-level auxiliary classifier outside the gradient, no building code	~	×	Synthetic (FEM)
Zhang et al. [15], GNN	RC frames	Inter-storey drift, lateral stiffness	Drift and stiffness limits as differentiable optimisation constraints, no specific code named	×	×	Synthetic (FEM)
Qin et al. [10], het. GNN	RC frames	Member sizes	Post-hoc drift check (GB 50011) outside the model	×	×	Real (95 buildings)
Jayaweera et al. [9], het. GNN	RC slab-beam floors	Actions, sizes, reinforcement	EN 1992-1-1 as penalised constraints in the member-sizing objective	×	×	Synthetic (FEM)
Verduzco et al. [11], PI-GNN	RC beams	Rebar area, layer count	Hong Kong code plus elasticity-PDE loss, auxiliary classifier predicts layer count, not compliance	~	×	Synthetic (optimisation)
Fei et al. [31], GraphSAGE-based	RC shear-wall buildings	Concrete and steel quantities, building-level	Minimum-reinforcement code rules embedded in loss and output layer, no per-element verdict	×	×	Real test set, GAN-augmented training
Schäfer et al. [12], GAT (prior work)	RC slab-column	Per-element compliance, reinforcement, two single-task models	Learned per-element compliance classification, plain cross-entropy	✓	×	Synthetic (FEM)
This work, GraphSAGE MTL	RC slab-column	Per-element compliance, reinforcement, shared encoder	Learned per-element compliance classification, cost-sensitive focal loss	✓	✓	Synthetic (FEM)

5.2. Operating Point and Element-Type Behaviour

The two error types carry asymmetric engineering consequences. A non-compliant element passed as compliant propagates undetected into detail design and constitutes a structural safety risk, while a compliant element flagged as non-compliant triggers only an unnecessary finite element verification. Because the two classes are well separated overall, with an ROC area-under-curve of 0.967 across the five seeds, the decision threshold can be shifted to set how conservative the screen is without sharply worsening either error type. Raising the threshold makes the screen more conservative, passing fewer elements as compliant and screening out more for verification, so that fewer non-compliant elements slip through at the cost of more compliant elements sent to an unnecessary check, while lowering it makes the screen more permissive in the opposite direction. The fixed decision threshold of 0.50 reported in this paper is one position on this ROC curve, and the choice of operating point is a designer decision illustrated by the predicted-score distributions in Figure 7. Two engineering dimensions inform that choice. Error tolerance depends on the design phase, with pre-design exploration tolerating more false alarms on the compliant class because rejected candidates can be revisited, while detail-design phases impose stronger efficiency constraints. The liability framing of the deployed tool sets the maximum tolerable column pass-through-rate, with a conservative screening filter requiring a rate close to zero and a more permissive design-hint tool tolerating higher

rates in exchange for higher recall on the compliant class. The asymmetric base rates of non-compliant slabs (approximately 29%) and non-compliant columns (approximately 2.5%) make element-specific thresholds defensible along both dimensions. Carried into the gradient-based design loop of Section 5.4, a conservative operating point would push essentially all non-compliant elements to larger cross-sections, giving a safer but heavier design, while a permissive point would favour leaner cross-sections and accept that more elements reach the downstream finite element check uncorrected.

The link between the classification and regression heads is itself element-type dependent. For columns the compliance label is the threshold function of a regressed quantity, the 4% reinforcement-ratio limit, so the classification head could in principle be replaced by a threshold on the regressed ratio. That substitution collapses the screen, as Section 4.4 reports. The two routes are not equivalent at the limit. A reinforcement estimate optimised for average accuracy is dominated by the many columns far below the 4% limit and tracks the rare heavily-reinforced columns near it poorly, whereas the compliance prediction targets the pass-or-fail decision directly and weights those rare critical columns heavily. Where the two routes disagree, the learned head acts as the strictly more conservative screen. For slabs the question does not arise, because the $l/250$ deflection label is not among the regression targets, so the slab compliance prediction is a genuinely independent learned signal.

Edge directionality contributes to this screening profile from the architectural side. The undirected variant reported in Section 4.4 raised the column pass-through-rate while leaving the regression error unchanged, which indicates that the directional inductive bias acts selectively on the column-screening axis. A comparable architectural lever is the choice of convolutional operator, where GraphSAGE under identical focal loss reaches a column pass-through-rate well below the GAT and GCN variants reported in Table 2, indicating that its neighbourhood aggregation is better matched to the screening task and motivating the operator selection.

A comparable conservatism-efficiency trade-off appears in graph-based sizing work in a different form. The fast graph-based sizing of Qin et al. [10] returns member sizes in under a second but emits neither reinforcement quantities nor compliance labels, and verifies the governing drift limit only post hoc through separate structural-analysis runs. SAGE_MTL instead produces compliance probabilities and reinforcement quantities jointly and exposes the conservatism-efficiency trade-off explicitly, with the operating point as the designer-controlled bridge between the two outputs.

Prediction behaviour is not uniform across the two element classes. Columns achieve higher classification accuracy than slabs. Two mechanisms account for this pattern. The graph representation captures column behaviour more consistently, with each column corresponding to a single node and a small set of adjacencies, while slab predictions aggregate local stress fields across many nodes and remain more sensitive to local load effects and support conditions. The asymmetric base rates of non-compliant slabs and columns reported in Section 4.1 place the discrimination burden disproportionately on the slab side, where roughly one in three elements is non-compliant against one in forty on the column side. As a consequence, column predictions are robust enough for screening at the default operating point, while slab predictions favour a more conservative threshold and benefit most from downstream FEM verification.

The regression provides a starting reinforcement estimate suited to the early-stage predimensioning role of the surrogate, which the subsequent detailed design refines to the final reinforcement.

5.3. Surrogate Scope and Verification Imperative

SAGE_MTL operates as a screening tool ahead of detailed structural verification. At the reported operating point the screen correctly flags approximately 94 out of every hundred non-compliant columns for verification, while the remaining six, a column pass-through-rate of 0.061, are predicted as compliant on average across the test set. This level is acceptable for narrowing the design space during exploration, while remaining incompatible with the deterministic safety guarantees expected from final structural design. The narrowing takes two deployment forms, a triage mode in which the surrogate screens many parametrically generated variants in milliseconds and forwards only the retained candidates to the slower finite element check, and a complementary iterative mode, left to the future work of Section 5.4, in which per-element gradients would refine one design before a final verification. In both forms the finite element check is reserved for the designs that survive the screen, so a false alarm costs only screening effort or a more conservative cross-section, while a missed non-compliance would enter the retained design unchecked. Verification through finite element analysis and the full Eurocode design check therefore remains a non-optional downstream stage after any optimisation step that uses the surrogate as a feedback signal. The role of the surrogate is to deliver fast feedback during iteration, while the role of finite element analysis is to confirm that the selected design satisfies all relevant safety, serviceability, and durability requirements.

The evidence presented here is bounded by the synthetic data source. The dataset comprises 2562 synthetic building graphs adapted from a single synthetic layout source [14], with structural systems restricted to slab-and-column reinforced concrete configurations. The training data also fixes the loading and material context, with a single dead and imposed load set, one quasi-permanent serviceability and one ultimate load combination, concrete grade C30/37, and exposure class XC1, so the surrogate has not encountered alternative load levels, concrete classes, or durability requirements. With the load case and material context held constant, compliance in this dataset varies primarily with geometry, namely span, slab thickness, and column cross-section, which delimits the regime the surrogate has learned and the claims the reported metrics can support. The structural analyses are linear-elastic and do not include creep, shrinkage, or long-term effects. Punching shear at the slab-column connection is likewise outside the modelled checks, although for flat slab-and-column systems it can govern the ultimate-limit-state design and constrain slab thickness from the predimensioning stage onward. The suitability of the surrogate for configurations in which punching shear governs is therefore not established by the present evidence, which keeps the downstream verification stage non-optional for such cases. External validation against real-project data or higher-fidelity nonlinear analyses is therefore required before the model is deployed in production workflows. Real building stock departs from the synthetic layouts along several axes the training distribution does not cover, including geometric irregularity, slab openings for stairwells, shafts, and cores, and mixed structural systems that combine slab-and-column framing with walls or bracing. The closed slab fields of the present dataset, which carry no openings, are a direct consequence of the layout source and mark the clearest of these distribution shifts. Because real project geometries can pass through the same automated finite element pipeline to obtain compliance and reinforcement labels, these cases can be added to the training distribution and the surrogate retrained on the broadened set. Closing the gap is therefore a question of data coverage that the existing method already accommodates. Deployment readiness would then require re-estimating the pass-through-rate on an independent, real-world labelled set and confirming that it stays within the tolerance set by the chosen operating point, since the rates reported here characterise the synthetic regime alone. The variability reported across the five seeds reflects training-initialisation randomness only, because the

data split and the input features were held fixed, so it does not capture the additional variance that resampling the split or the dataset would introduce.

Eurocode compliance is approximated by one criterion per element type, namely the $l/250$ deflection limit for slabs from EN 1992-1-1, §7.4.1 (4) and the 4% maximum reinforcement ratio for columns from EN 1992-1-1, §9.5.2 (3). These criteria were selected as representative compliance anchors for the early-stage predimensioning context. The full Eurocode 2 design check (ultimate-limit-state bending and shear, punching shear, durability detailing, fire resistance) is therefore not represented in the surrogate's compliance prediction. The operating point of 0.50 reported in this paper reflects an author-defined balance between conservative compliance screening and recall on the compliant class, and a designer with different cost asymmetries, for example a high computational verification cost downstream, may shift the threshold.

5.4. Future Research Directions

Operationalisation of the differentiable compliance signal within a closed gradient-based design loop (Figure 8) is the principal next step, complemented by an explainability-based design-change recommender that turns per-element compliance probabilities into actionable interventions. Dataset expansion towards richer synthetic boundary cases is an immediate parallel track.

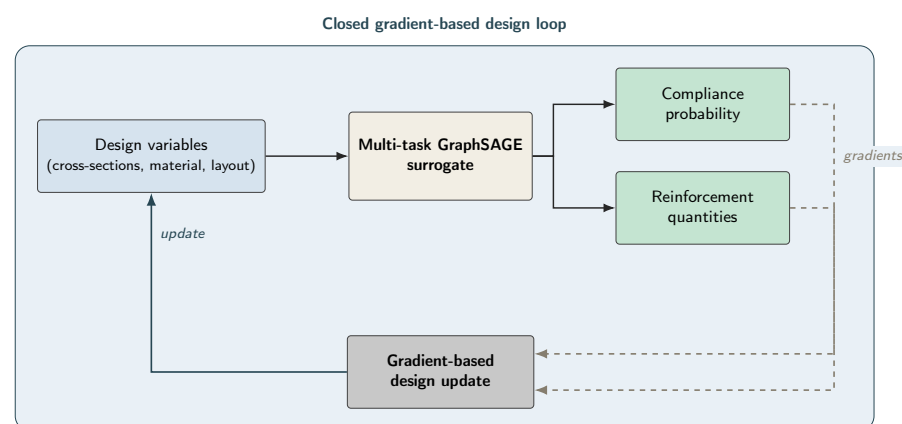


Figure 8. Proposed closed gradient-based design loop for the differentiable SAGE_MTL surrogate (future work). Each forward pass returns per-element compliance probabilities and reinforcement quantities, whose gradients feed an outer optimisation step that updates the design variables.

Per-element compliance probabilities provide a localised signal of which structural elements drive non-compliance, and that signal is the entry point for an explainability-based design-change recommender that identifies the subgraph pattern driving non-compliance, the candidate intervention most likely to restore compliance, and a ranked shortlist of design variants. Each surrogate forward pass returns predictions in milliseconds, so 10^4 design variants can be screened in seconds and produce a seed solution that a human engineer, a gradient-based or evolutionary outer optimiser, or a FEM verification stage can audit and refine.

Transfer learning across topologies and load cases has been shown to reduce the required number of finite element simulations by one to two orders of magnitude in graph-based truss-surrogate work [24], which suggests a viable route to broaden generalisability beyond the present synthetic dataset. Recent reviews of generative AI for structural design identify multi-constraint compliance, non-differentiable empirical rules, and limited data availability as field-wide bottlenecks [13], which motivates extension to heterogeneous element types, additional code bases, and trust-building tools for engineering adoption.

6. Conclusions

Early-stage predimensioning of reinforced concrete buildings requires rapid structural feedback that traditional finite element workflows cannot provide. This study presented a multi-task GraphSAGE surrogate model that produces such feedback for slab-and-column reinforced concrete buildings on the basis of 2562 building graphs derived from automated finite element simulations.

The contribution is threefold. The multi-task GraphSAGE architecture jointly predicts per-element Eurocode compliance and reinforcement quantities for slabs and columns through one end-to-end differentiable model. The cost-sensitive training regime and cost-sensitive reporting convention surface missed non-compliance that overall accuracy obscures in imbalanced compliance classification, combining pass-through-rate and balanced accuracy under focal-loss training. The operating-point selection framework makes the trade-off between conservative compliance screening and recall on the compliant class an explicit designer decision.

Across five random seeds, the model achieves a balanced accuracy of 90.9%, a column pass-through-rate of 6.1%, a slab pass-through-rate of 1.6%, and a mean reinforcement error near 2% of the normalised target range on held-out data, with an ROC area-under-curve of 0.97 and inference of approximately 0.5 ms per building graph.

In deployment the surrogate screens parametrically generated designs in milliseconds and forwards only the retained candidates to slower finite element verification, so the operating point becomes the lever through which a designer balances the tolerated risk of missed non-compliance against the cost of unnecessary checks. The same screen handles both the exploration of many distinct layout variants and the parametric optimisation of a single layout whose cross-sections are swept within a constrained design space.

The current evidence is bounded by the synthetic dataset, the linear-elastic structural representation, and a fixed load, material, and exposure context. Eurocode compliance is approximated by one criterion per element type, namely the $l/250$ deflection limit for slabs and the 4% maximum reinforcement ratio for columns, so the full Eurocode 2 design check is not represented in the surrogate's compliance prediction. Final structural verification through finite element analysis and the full Eurocode design check therefore remains a non-optional downstream stage before deployment in production workflows. Future work should operationalise the differentiable surrogate within a closed gradient-based design loop, develop the explainability-based design-change recommender outlined in Section 5.4, and extend the architecture to heterogeneous element types and additional code bases. Within these boundaries, the model shows that a differentiable, multi-task graph surrogate can deliver fast, cost-sensitive compliance screening for early-stage reinforced concrete predimensioning while preserving the gradient flow required for downstream design exploration.

Author Contributions: Conceptualization, N.S.; methodology, N.S.; software, N.S.; validation, N.S.; formal analysis, N.S.; investigation, N.S.; data curation, N.S.; writing—original draft preparation, N.S.; writing—review and editing, N.S., U.R. and J.D.; visualization, N.S.; supervision, U.R. and J.D. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) and the Open Access Publishing Fund of Technical University of Darmstadt.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used Claude (Opus 4.7, 4.8) from Anthropic, 2026, in order to improve language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GNN	Graph Neural Network
GCN	Graph Convolutional Network
GAT	Graph Attention Network
SAGE_MTL	Multi-Task GraphSAGE (this work)
SAGE_STL	Single-Task GraphSAGE (ablation baseline)
RC	Reinforced Concrete
FEM	Finite Element Method
RSA	Robot Structural Analysis
BCE	Binary Cross-Entropy
PTR	Pass-Through-Rate
ROC	Receiver Operating Characteristic

References

1. United Nations Environment Programme. *Not Just Another Brick in the Wall: The Solutions Exist-Scaling Them Will Build on Progress and Cut Emissions Fast. Global Status Report for Buildings and Construction 2024/2025*; United Nations Environment Programme: Nairobi, Kenya, 2025. <https://doi.org/10.59117/20.500.11822/47214>.
2. Thai, H.T. Machine Learning for Structural Engineering: A State-of-the-Art Review. *Structures* **2022**, *38*, 448–491. <https://doi.org/10.1016/j.istruc.2022.02.003>.
3. Samadian, D.; Muhit, I.B.; Dawood, N. Application of Data-Driven Surrogate Models in Structural Engineering: A Literature Review. *Arch. Comput. Methods Eng.* **2025**, *32*, 735–784. <https://doi.org/10.1007/s11831-024-10152-0>.
4. Jia, Y.; Wang, J.; Shou, W.; Hosseini, M.R.; Bai, Y. Graph Neural Networks for Construction Applications. *Autom. Constr.* **2023**, *154*, 104984. <https://doi.org/10.1016/j.autcon.2023.104984>.
5. Wettewa, S.; Hou, L.; Zhang, G. Graph Neural Networks for Building and Civil Infrastructure Operation and Maintenance Enhancement. *Adv. Eng. Inform.* **2024**, *62*, 102868. <https://doi.org/10.1016/j.aei.2024.102868>.
6. Liao, W.; Lu, X.; Huang, Y.; Zheng, Z.; Lin, Y. Automated Structural Design of Shear Wall Residential Buildings Using Generative Adversarial Networks. *Autom. Constr.* **2021**, *132*, 103931. <https://doi.org/10.1016/j.autcon.2021.103931>.
7. Zhao, P.; Liao, W.; Huang, Y.; Lu, X. Intelligent Design of Shear Wall Layout Based on Graph Neural Networks. *Adv. Eng. Inform.* **2023**, *55*, 101886. <https://doi.org/10.1016/j.aei.2023.101886>.
8. Li, M.; Liu, Y.; Wong, B.C.L.; Gan, V.J.L.; Cheng, J.C.P. Automated Structural Design Optimization of Steel Reinforcement Using Graph Neural Network and Exploratory Genetic Algorithms. *Autom. Constr.* **2023**, *146*, 104677. <https://doi.org/10.1016/j.autcon.2022.104677>.
9. Jayaweera, N.; Abeyrathna, A.; Herath, S.; Ekanayake, I.U.; Meddage, D.P.P. Code-Compliant Optimal Design of Reinforced Concrete Slab Beam Systems for Low to Mid-Rise Buildings Using Heterogeneous Graph Neural Networks with Attention Layers. *J. Build. Eng.* **2026**, *117*, 114599. <https://doi.org/10.1016/j.jobee.2025.114599>.
10. Qin, S.; Liao, W.; Huang, Y.; Zhang, S.; Gu, Y.; Han, J.; Lu, X. Intelligent Design for Component Size Generation in Reinforced Concrete Frame Structures Using Heterogeneous Graph Neural Networks. *Autom. Constr.* **2025**, *171*, 105967. <https://doi.org/10.1016/j.autcon.2025.105967>.
11. Verduzco, L.F.; Cheng, J.C.P.; Li, M.; Wang, L. Constructability-Aware Physics-Informed Graph Neural Networks for Surrogate-Assisted Optimization Design of Rebar in Concrete Beams. *Autom. Constr.* **2026**, *182*, 106760. <https://doi.org/10.1016/j.autcon.2026.106760>.
12. Schäfer, N.; Köhle, J.F.; Díaz, J. Data-Driven Predimensioning: Applying Graph Neural Networks to Reinforced Concrete Design. In *Proceedings of the Advances in Information Technology in Civil and Building Engineering*; Francis, A., Miresco, E., Melhado, S., Eds.; Springer: Cham, Switzerland, 2025; pp. 181–190. https://doi.org/10.1007/978-3-031-87364-5_15.
13. Liao, W.; Lu, X.; Fei, Y.; Gu, Y.; Huang, Y. Generative AI Design for Building Structures. *Autom. Constr.* **2024**, *157*, 105187. <https://doi.org/10.1016/j.autcon.2023.105187>.
14. Chang, K.H.; Cheng, C.Y. Learning to Simulate and Design for Structural Engineering. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2020; pp. 1426–1436.
15. Zhang, C.; Tao, M.x.; Wang, C.; Yang, C.; Fan, J.S. Differentiable Automatic Structural Optimization Using Graph Deep Learning. *Adv. Eng. Inform.* **2024**, *60*, 102363. <https://doi.org/10.1016/j.aei.2024.102363>.

16. Caruana, R. Multitask Learning. *Mach. Learn.* **1997**, *28*, 41–75. <https://doi.org/10.1023/A:1007379606734>.
17. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. <https://doi.org/10.2753/MIS0742-1222240302>.
18. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design Science in Information Systems Research. *MIS Q.* **2004**, *28*, 75–106.
19. 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* **2024**, *59*, 105712. <https://doi.org/10.1016/j.istruc.2023.105712>.
20. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907. <https://doi.org/10.48550/arXiv.1609.02907>.
21. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903. <https://doi.org/10.48550/arXiv.1710.10903>.
22. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*; NIPS: La Jolla, CA, USA, 2017; Volume 30.
23. Li, Y.; Wang, C.; Hou, W. A Universal Surrogate Modeling Method Based on Heterogeneous Graph Neural Network for Nonlinear Analysis. *Comput. Methods Appl. Mech. Eng.* **2025**, *437*, 117793. <https://doi.org/10.1016/j.cma.2025.117793>.
24. Whalen, E.; Mueller, C. Toward Reusable Surrogate Models: Graph-Based Transfer Learning on Trusses. *J. Mech. Des.* **2022**, *144*, 021704. <https://doi.org/10.1115/1.4052298>.
25. Nourian, N.; El-Badry, M.; Jamshidi, M. Design Optimization of Truss Structures Using a Graph Neural Network-Based Surrogate Model. *Algorithms* **2023**, *16*, 380. <https://doi.org/10.3390/a16080380>.
26. Tang, A.; Li, C.; Yang, J.; Zhang, H.; Zheng, Q.; Zhang, J. Training and Application of Graph Neural Networks for Predicting Structural Responses Targeted at Tall Building Structures. *J. Build. Eng.* **2025**, *103*, 112131. <https://doi.org/10.1016/j.jobe.2025.112131>.
27. Kraus, M.A.; Taras, A. Physik-informierte Künstliche Intelligenz zur Berechnung und Bemessung im Stahlbau. *Stahlbau* **2020**, *89*, 824–832. <https://doi.org/10.1002/stab.202000074>.
28. Fisch, R.; Stecker, E.; Kraus, M.A. Maschinelles Lernen beim Entwurf und der Bemessung von Stahlrahmenhallen. *Stahlbau* **2023**, *92*, 332–344. <https://doi.org/10.1002/stab.202200054>.
29. Bloch, T.; Borrmann, A.; Pauwels, P. Graph-Based Learning for Automated Code Checking—Exploring the Application of Graph Neural Networks for Design Review. *Adv. Eng. Inform.* **2023**, *58*, 102137. <https://doi.org/10.1016/j.aei.2023.102137>.
30. Alnuzha, M.; Bloch, T. The Role of Machine Learning in Automated Code Checking—A Systematic Literature Review. *J. Inf. Technol. Constr.* **2025**, *30*, 22–44. <https://doi.org/10.36680/j.itcon.2025.002>.
31. Fei, Y.; Liao, W.; Lu, X.; Guan, H. Knowledge-Enhanced Graph Neural Networks for Construction Material Quantity Estimation of Reinforced Concrete Buildings. *Comput.-Aided Civ. Infrastruct. Eng.* **2024**, *39*, 518–538. <https://doi.org/10.1111/mice.13094>.
32. Soledispa, C.E.; Pizarro, P.N.; Massone, L.M. Optimizing Reinforced Concrete Walls and Columns through Artificial Neural Networks with Structural Neighbor-Based Features. *J. Build. Eng.* **2024**, *89*, 109223. <https://doi.org/10.1016/j.jobe.2024.109223>.
33. Ali, S.; Ahmad, J.; Iqbal, U.; Khan, S.; Hadi, M.N.S. Neural Network-Based Models versus Empirical Models for the Prediction of Axial Load-Carrying Capacities of FRP-reinforced Circular Concrete Columns. *Struct. Concr.* **2024**, *25*, 1148–1164. <https://doi.org/10.1002/suco.202300420>.
34. Wang, H.; Zhang, P.; Cheng, H.; Hu, C.; Yu, B. Explicit Classification for Seismic Failure Modes of Corroded Reinforced Concrete Columns Using Class-weight-optimized Logistic Regression. *Struct. Concr.* **2026**, *suco.70599*. <https://doi.org/10.1002/suco.70599>.
35. Hu, M.; Fan, G.; Li, J.; Xia, Y.; Hao, H. Deep Learning Based Data Anomaly Detection Considering Data Imbalance and Extreme Loadings. *J. Build. Eng.* **2025**, *111*, 113632. <https://doi.org/10.1016/j.jobe.2025.113632>.
36. Fei, Y.; Qin, S.; Liao, W.; Guan, H.; Lu, X. Graph Neural Network-Assisted Evolutionary Algorithm for Rapid Optimization Design of Shear-Wall Structures. *Adv. Eng. Inform.* **2025**, *65*, 103129. <https://doi.org/10.1016/j.aei.2025.103129>.
37. Lou, H.; Gao, B.; Jin, F.; Wan, Y.; Wang, Y. Shear Wall Layout Optimization Strategy for High-Rise Buildings Based on Conceptual Design and Data-Driven Tabu Search. *Comput. Struct.* **2021**, *250*, 106546. <https://doi.org/10.1016/j.compstruc.2021.106546>.
38. Zhang, Y.; Wang, H.; Liu, J.; Liu, F.; Lv, X. Intelligent Structural Design of Composite Concrete-Encased Steel Columns Based on Hybrid Machine Learning and Multiobjective Optimization. *Struct. Concr.* **2026**, *27*, 49–72. <https://doi.org/10.1002/suco.70292>.
39. Kuhn, S.V.; Weber, M.; Binggeli, A.; Kraus, M.A.; Pérez-Cruz, F.; Kaufmann, W. Predictive Structural Assessment with Bayesian Deep Learning. *Adv. Eng. Inform.* **2025**, *68*, 103721. <https://doi.org/10.1016/j.aei.2025.103721>.
40. European Committee for Standardization (CEN). *Eurocode 2: Design of Concrete Structures—Part 1-1: General Rules and Rules for Buildings*; European Committee for Standardization (CEN): Brussels, Belgium, 2004.
41. Elkan, C. The Foundations of Cost-Sensitive Learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*; Lawrence Erlbaum Associates Ltd.: Mahwah, NJ, USA, 2001; Volume 17, pp. 973–978.

42. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. <https://doi.org/10.1109/TKDE.2008.239>.
43. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*; IEEE: New York, NY, USA, 2017; pp. 2980–2988.
44. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980v9. <https://doi.org/10.48550/arXiv.1412.6980>.
45. Ruder, S. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv* **2017**, arXiv:1706.05098. <https://doi.org/10.48550/arXiv.1706.05098>.

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.