Proceedings of IASS-IWSS 2026 — September 14-18, 2026, Turin, Italy
"R-Evolution of Shapes: Sustainability, Re-Use, and new Design Paradigms"
Annual Symposium of the International Association for Shell and Spatial Structures
3rd Italian Workshop on Shell and Spatial Structures
A. Manuello Bertetto, S. Gabriele, F. Marmo, A. Micheletti (eds.)

A Stock-Conditioned Generative Force Density Method Graph Machine Learning Framework for Structural Reuse

Yifan XIE*, [a], Lazlo BLEKER[b], Kam-Ming Mark TAM[a]

*, [a] The University of Hong Kong
StructurAI Lab, Department of Architecture, The University of Hong Kong, Hong Kong SAR, China
yifan_xie@connect.hku.hk

[b] Technical University of Munich

Abstract

This paper presents a stock-conditioned generative form-finding framework for structural reuse via materially efficient funicular design. Rather than treating reuse as the matching of reclaimed materials to a predefined design, the framework allows the design and its problem formulation, such as boundary and support conditions, to adapt to available stock. This is important because a fixed design problem may itself limit effective reuse. Current approaches do not jointly integrate structural form-finding, design adaptation, demand-to-stock assignment, and multiple-solution generation. The proposed framework brings these capabilities together within a unified learning process.

Specifically, the framework combines graph neural networks (GNNs), differentiable Force Density Method (FDM) form-finding, and optimal transport (OT)-based demand-to-stock matching in a generative pipeline trained across stock distributions, geometries, and topologies. Conditioned on available stock and latent design variables, the GNN predicts force densities and boundary changes. Differentiable FDM maps these predictions to efficient funicular structures, while OT guides stock-to-demand assignment towards reuse of reclaimed elements. Latent-space variation enables multiple designs under the same stock and initial conditions.

Preliminary results show feasible alternatives with improved reuse rates relative to optimisation-based baselines spanning different degrees of design flexibility. The trained model reduces computation time from 30,000 ms to 4 ms per design across unseen scenarios. Together, these results demonstrate the potential of differentiable physics-informed generation to learn across coupled objectives of structural efficiency, material reuse, and design adaptation, reframing structural reuse from post-design material assignment as a stock-conditioned design paradigm in which available materials shape the design itself.

Keywords: Structural Reuse; Stock-Conditioned Design; Force Density Method; Differentiable Form-Finding; Graph Neural Networks; Optimal Transport; Generative Design

1. Introduction

1.1. Toward a New Design Paradigm for Structural Reuse

The buildings and construction sector accounts for approximately 34% of global CO_2 emissions, with structural materials contributing significantly to this total [1]. This makes structural reuse a necessary strategy for reducing embodied carbon and material waste. Unlike conventional structural design, in which components are manufactured after the structure has been defined, reuse-oriented design requires the structure itself to be assembled from a given stock of available elements. To enable effective reuse, it is important for design methods to move beyond operating on fixed geometries and instead allow structural form and design intent to adapt to the available stock, as otherwise the space of feasible solutions may be unnecessarily constrained. Accordingly, computational tools should support both the alignment of structural demand with available stock and the exploration of multiple feasible design possibilities. Existing approaches, however, remain limited in this regard. The present study therefore goes further by treating available stock not as a constraint on a predefined brief, but as a driver of structural form generation, and proposes a generative framework that enables the design, including its boundary conditions, to evolve in response to available stock while supporting multiple feasible solutions, as illustrated in the output and paradigm rows of Fig. 1(c).

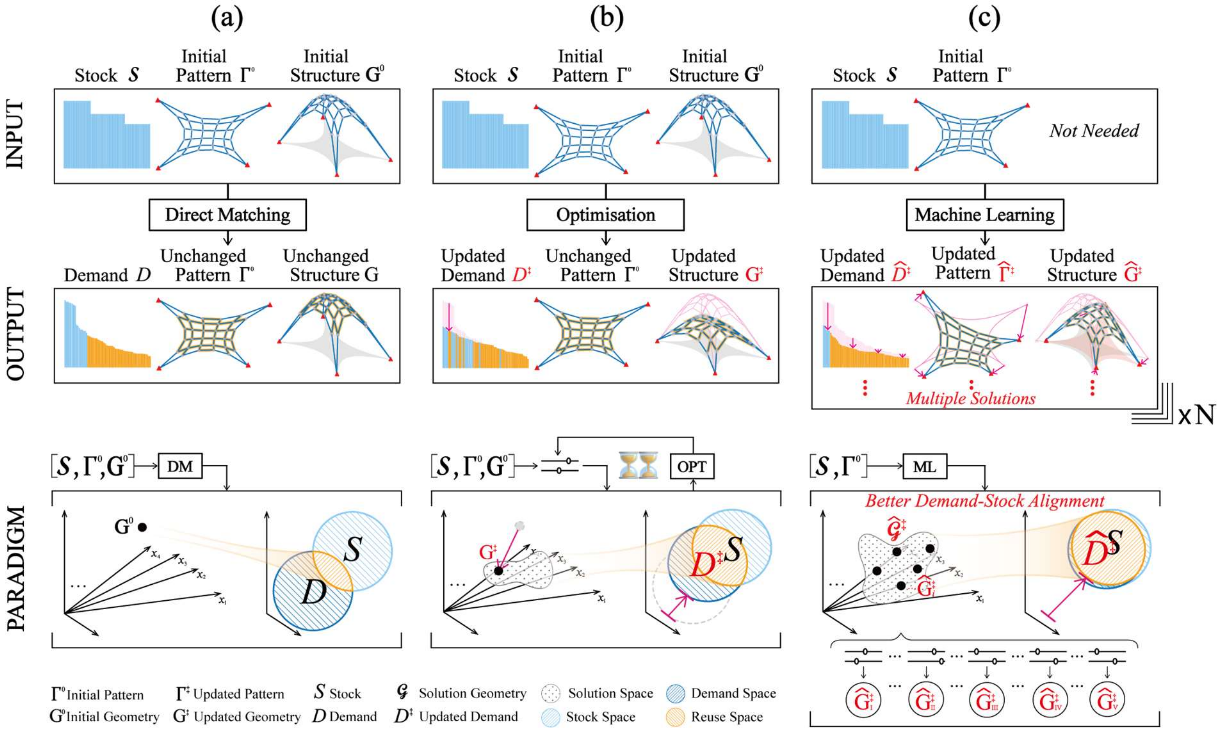

Figure 1: Conceptual comparison of reuse-oriented design workflows. The three columns represent (a) stock-constrained direct matching, (b) differentiable stock-driven optimisation, and (c) the proposed learning-based generative framework. The three rows illustrate the corresponding input, output, and underlying paradigm, respectively. Direct matching assigns stock to a largely fixed target geometry; optimisation adjusts the design, or more specifically the demand space, to better align with the available inventory; and the proposed framework treats available stock as a driver of structural form generation, enabling the design, including its boundary conditions, to evolve and thereby expanding both the design and demand spaces while producing multiple feasible solutions. Each blue bar represents an individual member, while an orange overlay indicates a successful stock–demand match and hence reuse.

1.2. Related Work

Reuse-oriented computational design has primarily relied on optimisation, heuristic, and metaheuristic methods [2]. Exact approaches, such as Mixed Integer Linear Programming [3], [4] and assignment-based methods including the Hungarian algorithm [5], can explicitly formulate or solve stock assignment under structural and geometric constraints. Heuristic and metaheuristic methods, such as Best-Fit heuristics [6] and evolutionary algorithms [7], provide increased flexibility and computational efficiency, typically at the expense of reuse optimality. Despite their differences, these approaches operate within a stock-constrained workflow prescribed to a fixed target geometry, as conceptually illustrated in Fig. 1(a). This limitation extends to hybrid formulations such as Favilli et al. [8], in which differentiable geometric optimisation is introduced but remains coupled to a discrete inventory assignment process that constrains the space of feasible design solutions. As a result, the potential for jointly optimising geometry and stock assignment to reveal designs whose material demand better aligns with the available stock remains limited; the extent of achievable reuse is constrained by prior assignment decisions.

Recent work combining automatic differentiation, machine learning, and form-finding methods such as the Force Density Method (FDM) shows how differentiable programming enables end-to-end optimisation pipelines in which multiple design considerations can be integrated and jointly optimised within a single workflow [9], [10], [11],[12]. More specifically, in the context of structural reuse, Lee et al. [13], [14] propose the use of optimal transport as a differentiable surrogate for the stock assignment problem, relaxing discrete matching into a continuous formulation. This allows stock–demand compatibility to guide the optimisation of continuous design variables while deferring discrete assignment to post-processing, as illustrated in Fig.1(b). As a result, geometry and stock assignment can be explored in a more coupled manner, enabling the discovery of designs whose material demand better aligns with the available stock, rather than being constrained by prior assignment decisions. The integration of form-finding ensures that generated solutions follow efficient axial structural logic.

1.3. Research Gap and Objectives

Despite recent advances, several limitations remain in reuse-oriented computational design. This work addresses these limitations through a unified, differentiable, physics-informed graph neural network framework for structural reuse, in which FDM-based form-finding is embedded as a generative solver within the learning process. Its main contributions are as follows:

- ***Improved adaptability:*** The framework learns a conditional mapping from design inputs and available stock to feasible structural configurations across diverse boundary conditions, topologies, and inventory distributions. As a result, the trained model generalises to new design problems and inventory conditions without requiring repeated optimisation.
- ***Unified integration:*** It jointly optimises stock assignment, structural form generation, and boundary conditions of the design within a single differentiable pipeline, rather than treating them as separate or sequential stages. This enables the framework to better align the material demand of a design with the available stock.
- ***Multi-solution generation:*** It produces multiple feasible structural equilibria conditioned on the available inventory through a latent-space generative approach, with diversity regularisation encouraging geometrically distinct solutions. This enables the generation of diverse design alternatives rather than a single final output, thereby supporting exploratory design.

2. Methodology

2.1. Framework Overview

Given an initial pattern and available stock, the framework generates edge-based force densities that, when used within the FDM, produce equilibrium configurations that maximise reuse of the available stock. By varying a latent input, users can explore multiple alternative solutions under the same conditions, rather than converging to a single output. More concretely, the proposed differentiable, physics-integrated learning framework consists of three main components:

(1) Dataset generation: A dataset of diverse pattern–stock pairings is constructed to enable the model to learn across a wide range of stock-reuse design scenarios (Section 2.2).

(2) Graph-based inverse model: A graph-based surrogate model encodes the input pattern and available stock, and predicts edge-based force densities and support displacements for the target structure, conditioned on the stock distribution and a latent input (Section 2.3).

(3) Differentiable objectives and physics integration: The predicted displacements update the boundary conditions, and together with the predicted force densities, prescribed connectivity, and external loads, are passed to a FDM module to generate an equilibrium geometry. From this, demand members are extracted and compared to the available stock via a Sinkhorn-based objective. Additional regularisation encourages the generation of distinct yet feasible design alternatives under different latent inputs. The resulting loss is backpropagated to update the model parameters (Section 2.4).

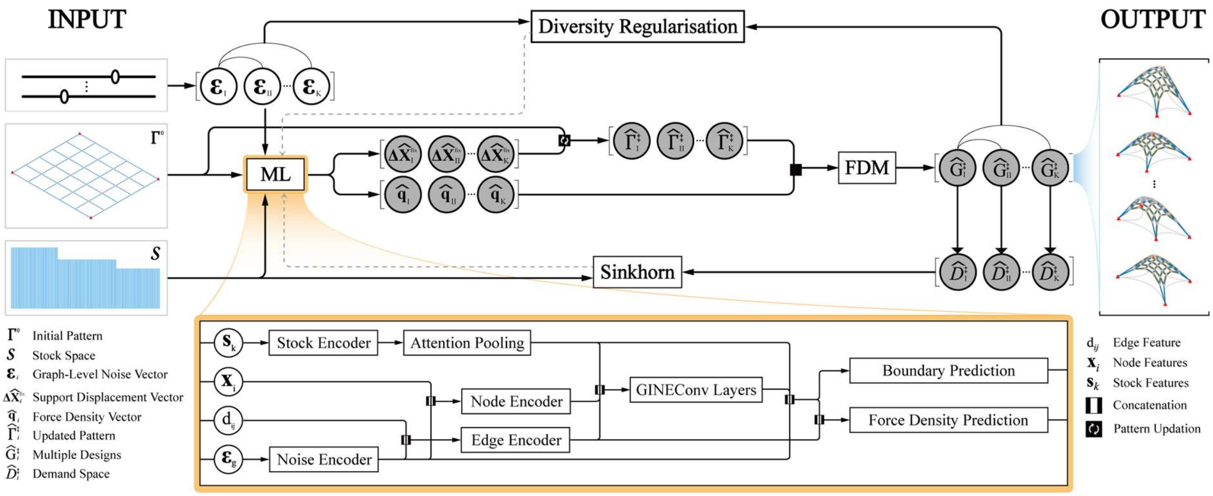

Figure 2: Overview of the proposed framework. Given an initial pattern, stock members, and a graph-level noise vector, the surrogate model predicts support displacement and force density, which are passed to a FDM module to generate equilibrium geometries and demand spaces. These are matched to the available stock through a Sinkhorn-based objective, while a distance regulariser promotes solution diversity. The lower panel shows the surrogate model architecture.

2.2. Training Data Generation

To support broader exploration of structural form and to better approximate realistic stock distributions, the training dataset was synthetically generated through a parametric simulation pipeline. On the demand side, boundary dimensions, pattern resolutions, support conditions, topologies, and geometric perturbations were systematically varied, with optional diagonal bracing, non-uniform node spacing, and curved boundary transformations in both in-plane and out-of-plane directions. Based on these settings, an initial pattern of the demand structure was constructed and used as input to the surrogate (see Section 2.4).

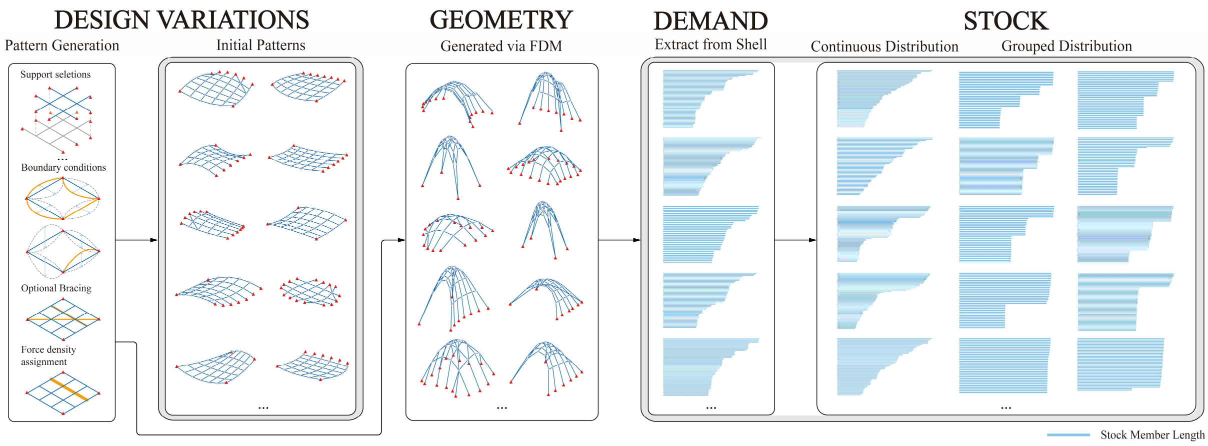

Figure 3: Generation pipeline of demand patterns and stock-members.

On the stock side, the objective was not to preserve an overall structural geometry, but to generate realistic member-level distributions. Accordingly, stock-side geometries were generated independently within the same framework and then reduced to stock members represented only by their lengths and cross-sectional areas. Global or local force density values were assigned, together with vertical loads applied at free nodes, and the FDM was used to obtain equilibrium geometries. These were then converted into stock members, with the required cross-sectional area of each member derived from its axial force under a prescribed allowable stress.

To better reflect realistic stock distributions, two inventory types were considered: one preserving the continuous simulated distribution, and the other regrouping members into clustered distributions with slight intra-group variation. The latter represents the repetitive yet non-identical member patterns often observed in reclaimed stock inventories (see Fig. 3).

2.3. Surrogate Modelling with Graphs

This section presents a graph-based surrogate model that predicts force density values for physical members and displacement vectors for fixed support nodes in grid-shell structures under static loading.

2.3.1. Encoding Pattern Configurations as Graphs

The initial demand configuration is encoded as a graph, where the set of vertices corresponds to joints and the set of edges corresponds to bars. Each vertex is assigned a feature vector consisting of its spatial coordinates and a binary support indicator. Each physical edge is associated with a scalar edge feature representing the original bar length.

2.3.2. Encoding Stock Members

In addition to the demand graph, the available stock is represented as a set of reusable members,

$$S = \{\mathbf{s_k}\}_{k=1}^{N_s}, \qquad \mathbf{s_k} = [L_k, A_k]^\top \in \mathbb{R}^2, \qquad \mathbf{X}^{\text{stock}} = \left[\mathbf{s_1}, \dots, \mathbf{s_{N_s}}\right]^\top \in \mathbb{R}^{N_\text{s} \times 2}, \tag{1}$$

where N_s denotes the number of stock members, and L_k and A_k denote the length and cross-sectional area of stock member k, respectively.

2.3.3. Graph Neural Network Architecture

The demand graph G, the stock matrix $\mathbf{X}^{\text{stock}}$, and a graph-level random noise vector $\boldsymbol{\varepsilon}_\text{g}$ are passed to a stock-aware graph neural network based on the Graph Isomorphism Network with Edge features (GINE) [15]. The architecture of this surrogate model is illustrated in the lower panel of Fig. 2. The network produces two outputs simultaneously: predicted force density values for edge members and displacement vectors for support. For each graph, a two-dimensional random vector $\boldsymbol{\varepsilon}_\text{g} \in \mathbb{R}^2$ is first sampled from a uniform distribution over $[-1,1]^2$ and encoded by a multilayer perceptron (MLP) into a latent vector:

$$\mathbf{z}_\text{g} = \phi_{\text{noise}}\left(\boldsymbol{\varepsilon}_\text{g}\right), \qquad \mathbf{z}_\text{g} \in \mathbb{R}^{D_z}. \tag{2}$$

This latent vector acts as a graph-level conditioning variable and is broadcast to all nodes and edges, enabling the model to generate multiple feasible solutions under identical geometric and stock conditions.

Node features $\mathbf{x}_i$ concatenated with $\mathbf{z}_\text{g}$ are passed through a node encoder MLP to produce initial node embeddings. Directed edge features, formed by concatenating d_{ij} with $\mathbf{z}_\text{g}$, are passed through an edge encoder MLP to obtain encoded edge attributes. Stock member features are embedded by an MLP and aggregated through an attention-based pooling module to obtain a graph-level stock embedding $\mathbf{h}^{\text{stock}}$, which is broadcast to the nodes and directed edges of the corresponding graph.

The resulting node embeddings and edge attributes are then processed through four GINE convolution layers to capture structural dependencies conditioned on geometry, supports, stock availability, and latent design variation. The resulting node representations are denoted by $\mathbf{h}_i$.

After message passing, each directed edge (i, j) is assigned an edge-level feature vector formed by concatenating the hidden representations of its incident nodes, the encoded edge feature, the pooled stock embedding, and the latent noise embedding. This feature is passed through an edge prediction head to produce a directed force density prediction:

$$\hat{\mathbf{q}}_{ij}^{\mathrm{dir}} = \phi_{\text{edge-head}}\big([\,\mathbf{h}_i \,\|\, \mathbf{h}_j \,\|\, \mathbf{e}_{ij} \,\|\, \mathbf{h}^{\mathrm{stock}} \,\|\, \mathbf{z}_{\mathrm{g}}]\big) \tag{3}$$

In parallel, a node-level prediction head maps the final node embedding, pooled stock embedding, and latent vector to a three-dimensional displacement vector:

$$\Delta\hat{\mathbf{x}}_i = \phi_{\text{support-head}}\big([\,\mathbf{h}_i \,\|\, \mathbf{h}^{\mathrm{stock}} \,\|\, \mathbf{z}_{\mathrm{g}}]\big) \tag{4}$$

The displacement branch is applied only to fixed support nodes. The raw displacements are bounded and scaled by a graph-wise radius limit, and Laplacian smoothing is applied over the fixed nodes to enforce local consistency. The resulting displacement field updates the support boundary conditions for the subsequent differentiable force density analysis. The complete surrogate model can therefore be written as:

$$\big(\hat{\mathbf{q}}, \Delta\hat{\mathbf{X}}^{\mathrm{fix}}\big) = f_\theta\big(\mathrm{G}, \mathbf{X}^{\mathrm{stock}}, \boldsymbol{\varepsilon}_{\mathrm{g}}\big), \tag{5}$$

where G denotes the demand graph, $\mathbf{X}^{\mathrm{stock}}$ the stock matrix, $\boldsymbol{\varepsilon}_{\mathrm{g}}$ the graph-level noise input, $\hat{\mathbf{q}}$ the predicted force density vector for the physical members, and $\Delta\hat{\mathbf{X}}^{\mathrm{fix}}$ the predicted displacement field of the fixed support nodes.

2.4. Physics-Integrated Learning Framework

This section presents a unified differentiable learning pipeline consisting of three components: equilibrium geometry generation, stock-demand matching, and diversity regularization. Together, these components allow the generated geometry and its resulting demand space to adapt to the available stock distribution, while supporting the generation of multiple valid design alternatives.

2.4.1. Form-Finding via the Force Density Method

The predicted force density values, together with the updated fixed-node coordinates, graph connectivity, and applied loads, are passed to a differentiable FDM module. Given the predicted force densities and boundary conditions, the FDM solves the equilibrium geometry by updating the free-node coordinates. From the resulting geometry, the demand members are extracted as:

$$D(\hat{\mathbf{q}}) = \{(L_i(\hat{\mathbf{q}}), A_i(\hat{\mathbf{q}}))\}_{i=1}^{N_{\mathrm{d}}} \tag{6}$$

where L_i is the deformed length of member i, and A_i is its required cross-sectional area. Since the overall pipeline is differentiable, gradients from downstream objectives can be propagated through the FDM step to update the predicted force densities and support displacements.

2.4.2. Stock-Demand Matching via Optimal Transport

A differentiable matching objective is defined between generated demand members and available stock members in a feature space of member length and cross-sectional area. Given the demand set D and stock set S, a pairwise cost matrix $\mathbf{C}$ is constructed, with an asymmetric cost that distinguishes insufficiency from cutoff. Insufficiency penalizes unmet length or cross-sectional requirements, whereas cutoff accounts for excess length, trimming waste, and oversupply of sectional capacity. The stock-demand matching problem is then formulated as an entropy-regularized unbalanced optimal transport problem:

$$\mathbf{P}^*(\hat{\mathbf{q}}) = \arg\min_{\mathbf{P}\geq 0}\{\langle \mathbf{P}, \mathbf{C}(\hat{\mathbf{q}})\rangle - \varepsilon H(\mathbf{P}) + \tau_{\mathrm{a}}\,\mathrm{KL}(\mathbf{P1} \parallel \mathbf{a}) + \tau_{\mathrm{b}}\,\mathrm{KL}(\mathbf{P}^{\top}\mathbf{1} \parallel \mathbf{b})\}, \tag{7}$$

where $H(\mathbf{P})$ is the entropy term, $\mathbf{a}$ and $\mathbf{b}$ are the reference demand and stock marginals, and $\tau_{\mathrm{a}}, \tau_{\mathrm{b}}$ control the strength of marginal matching on the demand and stock sides, respectively.

The corresponding optimal transport loss is written as:

$$\mathcal{L}_{\mathrm{OT}} = \langle \mathbf{P}^*, \mathbf{C}(\hat{\mathbf{q}})\rangle - \varepsilon H(\mathbf{P}^*) + \tau_{\mathrm{a}}\,\mathrm{KL}(\mathbf{P}^*\mathbf{1} \parallel \mathbf{a}) + \tau_{\mathrm{b}}\,\mathrm{KL}((\mathbf{P}^*)^{\top}\mathbf{1} \parallel \mathbf{b}) \tag{8}$$

An auxiliary loss $\mathcal{L}_{\mathrm{aux}}$ is further introduced to penalize excessive member length, required cross-sectional area, and overall generated height.

2.4.3. Distance Regularization for Multi-Solution Generation

To enable controllable multi-solution generation from the latent space, different graph-level noise samples are introduced during training, leading to different predicted force densities, support displacements, and equilibrium geometries. A distance regularization term is then applied to the fixed-support displacement fields to relate pairwise distances in latent space to corresponding physical distances. Latent distances are normalized by a graph-specific measure of the effective extent of the noise space, while physical distances are computed from mean-centered fixed-support displacement differences using an anisotropic metric and further normalized by a graph-specific geometric reference. The resulting regularization term for graph g is written as:

$$\mathcal{L}_{\mathrm{dist}} = \frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}}\left(d_{\mathrm{g,f}}^{(i,j)} - \beta d_{\mathrm{g,z}}^{(i,j)}\right)^2, \tag{9}$$

where $\mathcal{P}$ denotes the set of unordered pairs of noise samples, $d_{\mathrm{g,z}}^{(i,j)}$ is the normalized latent-space distance, $d_{\mathrm{g,f}}^{(i,j)}$ is the normalized physical distance, and β is a scaling parameter.

Averaging the distance regularization over all graphs in the batch gives L_{dist}, and the final objective is defined as:

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{OT}}\mathcal{L}_{\mathrm{OT}} + \lambda_{\mathrm{aux}}\mathcal{L}_{\mathrm{aux}} + \lambda_{\mathrm{dist}}\mathcal{L}_{\mathrm{dist}} \tag{10}$$

3. Results

3.1 Matching Performance

Matching performance was evaluated on the test set using reuse rate, defined as the proportion of demand members that can be feasibly matched to the available stock. Fig. 4 compares the reuse-rate distributions of the proposed framework, in both the with-support-change and without-support-change settings, against the optimisation-based method and the Direct-Matching baseline. The distributions of the proposed method are clearly shifted toward higher reuse rates, particularly in the support-change setting. Among all approaches, the proposed method with support change achieves the highest mean reuse rate at 93.65%, followed by the proposed method without support change at 87.23%, the optimisation-based method at 81.63%, and Direct-Matching at 68.78%. These results indicate that the proposed framework achieves better stock-demand matching and overall reuse performance across the test set.

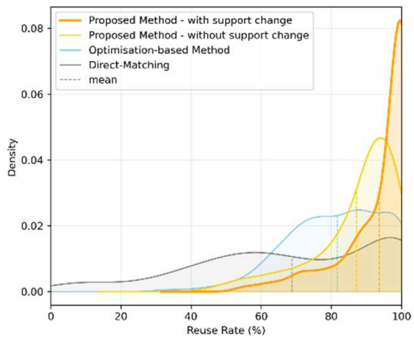

Figure 4: Reuse rate distribution of the proposed method and baseline approaches.

3.2 Computational Efficiency

Fig. 5 compares the four methods in terms of computation time and representative outputs. Direct-Matching is the fastest because it operates on a predefined geometry and performs only member assignment, but its reuse performance is limited by that constraint. The optimisation-based method improves stock–demand matching through iterative geometric adaptation, but at substantially higher runtime. By contrast, the proposed framework generates stock-conditioned solutions directly at inference time, the support-change variant further improves performance, achieving higher reuse rates with much lower runtime than the optimisation-based baseline, while also supporting multi-solution generation under the same input–stock condition.

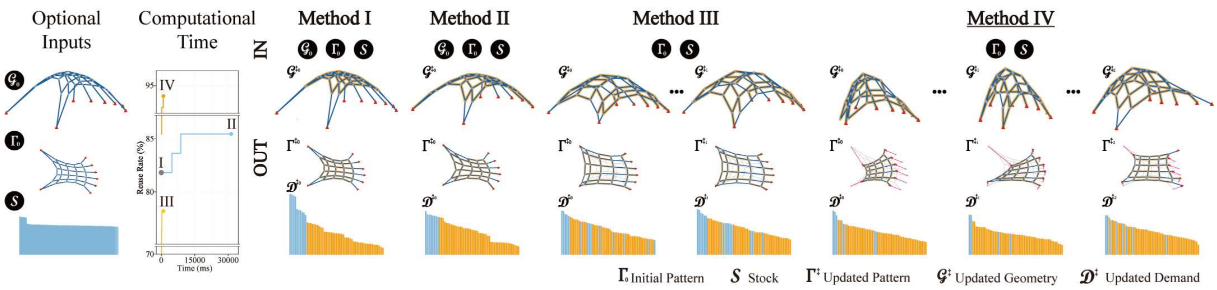

Figure 5: Comparison of computation time and representative outputs across four methods: Method I, Direct-Matching; Method II, the optimisation-based method; Method III, the proposed method without support change; and Method IV, the proposed method with support change.

3.3 Solution Diversity

Fig. 6 demonstrates the solution diversity enabled by the proposed framework. Under a fixed pattern–stock condition, varying the latent input produces multiple feasible structural equilibria with different geometries, boundary conditions, demand-member distributions, and reuse rates. Rather than converging to a single solution, the framework supports one-to-many generation and enables exploration of a broader stock-conditioned design space. This shifts reuse-oriented structural design from single-solution prediction toward controllable generation of multiple feasible alternatives.

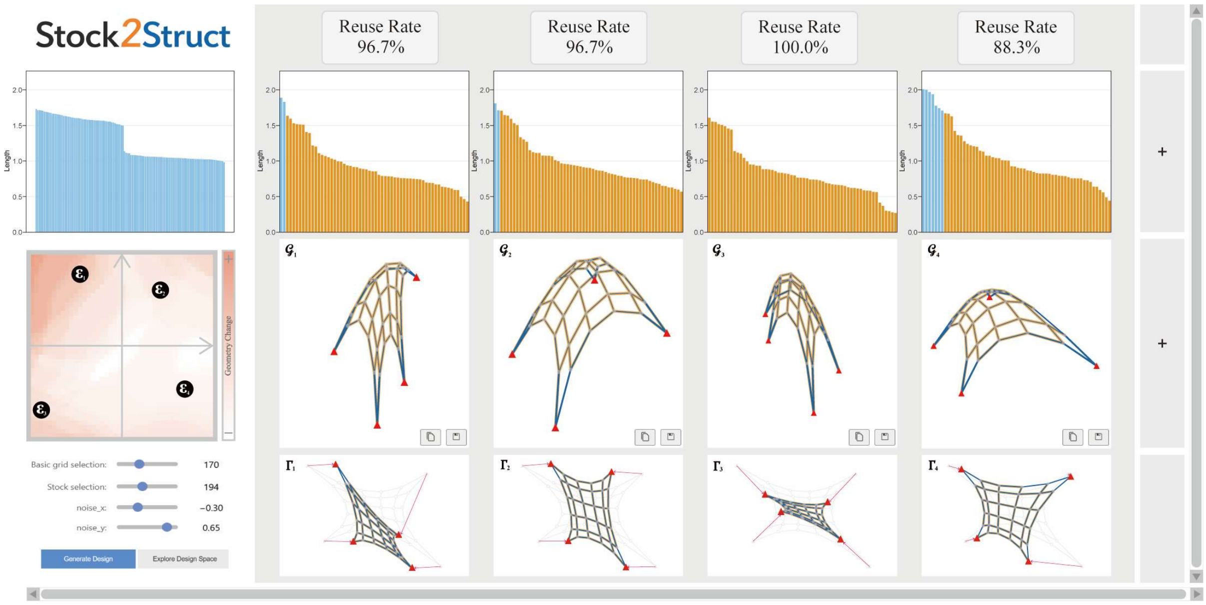

Figure 6: Design exploration of the multi-solution space generated by the proposed framework under a fixed pattern–stock condition, illustrating multiple feasible stock-conditioned solutions through controlled latent-space variation.

3.4 Generalisability

Fig. 7 further demonstrates that the proposed framework learns across diverse pairings of design inputs and available stock. For unseen combinations of patterns, topologies, support conditions, and stock sets, the model still generates feasible stock-conditioned structural solutions with relatively high reuse rates. This suggests that the framework captures transferable relationships between structural demand and stock availability, rather than relying on case-by-case optimisation or merely memorising specific training instances.

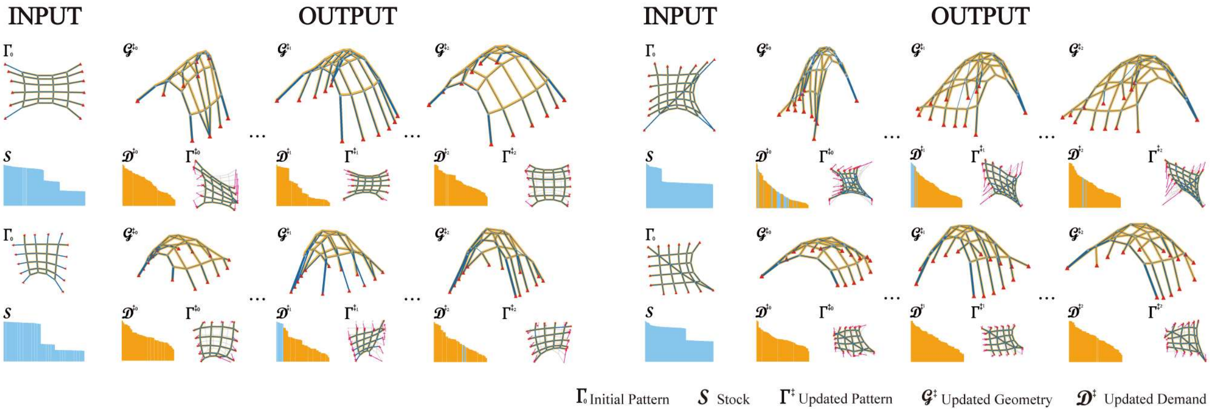

Figure 7: Examples demonstrating the generalisability of the proposed framework across unseen pairings of design inputs and available stock.

4. Conclusion

4.1 Limitations and Future Work

This study has several limitations. First, some generated cases exhibit overlapping or overly close supports, suggesting that support status and, more broadly, topology could be incorporated as design variables so that boundary conditions and connectivity are not fixed in advance. Second, Laplacian smoothing is currently applied only after support displacement prediction; integrating this regularisation directly into training could improve local consistency and enable smoothness to be learned jointly with boundary adaptation. Third, the current framework is limited to compression-only structures by constraining force densities to a single sign. More flexible formulations of force-density prediction could enable a broader range of structural behaviours and equilibrium geometries. Fourth, the stock representation remains simplified and does not yet capture the irregular geometries and structural behaviours of real reclaimed members, including variations in compression performance. Addressing these limitations could support a more open-ended generative framework in which structural layout, boundary conditions, and material availability are explored more jointly under realistic reuse constraints.

4.2 Summary of Contributions

This paper presented a differentiable, physics-informed, solver-integrated graph neural network framework for reuse-oriented structural design. By learning across diverse design–stock pairings, it links stock assignment, structural form generation, and the evolution of geometry, boundary conditions, and demand space within a single differentiable framework, reducing the need for case-by-case optimisation. The support-displacement variant further expands the feasible design space through boundary adaptation, while latent-space variation enables multiple feasible structural equilibria under the same inventory conditions. Overall, the results demonstrate improved stock-demand compatibility, higher reuse performance, and a more efficient and flexible workflow than Direct-Matching and the optimisation-based baseline.

Taken together, these results suggest a broader shift in reuse-oriented structural design: rather than treating stock as a constraint on a predefined brief, it can act as a driver of structural form generation, allowing form to adapt to available materials and emerge through reuse rather than being prescribed.

References

[1] United Nations Environment Programme, *2024/2025 Global Status Report for Buildings and Construction*, 2025. [Online]. Available: https://www.unep.org/resources/report/global-status-report-buildings-and-construction-20242025. [Accessed: Mar. 27, 2026].

[2] Q. Sun, D. Reisach, S. Langenberg and B. Dillenburger, "Computational design methods for geometry-driven upcycling of found objects in construction," *Automation in Construction*, vol. 183, Art. no. 106803, 2026.

[3] J. Brütting, J. Desruelle, G. Senatore and C. Fivet, "Design of truss structures through reuse," *Structures*, vol. 18, pp. 128–137, 2019.

[4] J. Warmuth, P. D'Acunto and C. Fivet, "Combining interactive equilibrium design and stock-constrained optimisation for component reuse," *Structures*, vol. 85, Art. no. 111095, 2026.

[5] Y. Huang, L. Alkhayat, C. De Wolf and C. Mueller, "Algorithmic circular design with reused structural elements: Method and tool," in *Proceedings of the International fib Symposium on the Conceptual Design of Structures*, Attisholz Areal, 2021, pp. 457–468.

[6] J. Brütting, P. O. Ohlbrock, J. Hofer and P. D'Acunto, "Stock-constrained truss design exploration through combinatorial equilibrium modeling," *International Journal of Space Structures*, vol. 36, no. 4, pp. 253–269, 2021.

[7] A. Van Marcke, V. Laghi and J. V. Carstensen, "Automated planar truss design with reclaimed partially disassembled steel truss components," *Journal of Building Engineering*, vol. 84, Art. no. 108458, 2024.

[8] A. Favilli, F. Laccone, P. Cignoni, L. Malomo and D. Giorgi, "Optimizing free-form grid shells with reclaimed elements under inventory constraints," *Computer Graphics Forum*, 2025.

[9] K.-M. M. Tam, P. D'Acunto, N. C. Brown, L. Bleker, T. Van Mele and P. Block, "A unified differentiable geometric deep learning message passing framework for inverse structural form-finding," *SSRN*, preprint, 2025, doi: 10.2139/ssrn.5316084.

[10] L. Bleker, P. D'Acunto and K.-M. M. Tam, "Generalized inverse structural form-finding using E(3)-equivariant physics-based graph neural networks," *SSRN*, preprint, 2025, doi: 10.2139/ssrn.5208576.

[11] R. Pastrana, E. Medina, I. M. de Oliveira, S. Adriaenssens and R. P. Adams, "Real-time design of architectural structures with differentiable mechanics and neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[12] K.-M. M. Tam, N. C. Brown, M. Bronstein, T. Van Mele and P. Block, "Learning constrained static equilibrium for thrust network inverse form-finding via physics-informed geometric deep learning on CW complexes," *Advanced Engineering Informatics*, vol. 70, Art. no. 103870, 2026.

[13] K. J. Lee, C. T. Mueller and Y. Huang, "Differentiable assignment for circularity-driven structural design and optimization," poster, *ACM SIGGRAPH Symposium on Geometry Processing*, 2024.

[14] K. J. Lee, "Geometric Interpretations of Structural Demand for the Analysis and Reduction of Design Complexity," Ph.D. dissertation, Department of Architecture, Massachusetts Institute of Technology, Cambridge, MA, 2025, ch. 6, pp. 159–175.

[15] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande and J. Leskovec, "Strategies for pre-training graph neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.