Hierarchical Multi-Level 3D Geometry Generation with Stress-Aware Learning

Anonymous authors
Paper under double-blind review

Abstract

Current approaches for Lego 3d structural assembly are usually learned to maximize IOU between generated output and target construction. We propose a new approach which is able to build stable structures based on physics-aware reward. Our method employs a two-level agent architecture in which a high-level PPO-based planner proposes a scheme, while a low-level Wave Function Collapse (WFC) agent handles precise brick placement with constraint satisfaction. Experimental results demonstrate that our hierarchical method consistently constructs structurally sound buildings while reducing material usage. We also show that replacing the computationally expensive FEM solver with fast FNO achieves comparable performance, confirming the approach's scalability for large-scale problems.

1 Introduction

The construction of 3D structures from discrete building elements represents a fundamental challenge in robotics, automated manufacturing, and architectural design. Traditional approaches to this problem rely heavily on topology optimization methods that require extensive computational resources and often produce solutions that are difficult to construct in practice. Recent progress, particularly with generative models and reinforcement learning, has enabled the creation of complex shapes and assemblies (1; 2). However, a significant gap remains between generating objects that are visually plausible and those that are physically viable. Most current approaches to LEGO brick construction focus on replicating a target shape, commonly employing geometric metrics such as Intersection over Union (IoU) as the main optimization criterion (1). Although effective within their intended scope, these methods do not ensure that the resulting structure can support its own weight or resist external forces. We focus on a more demanding and application-driven challenge: structurally-informed combinatorial construction. The objective moves past simply assembling bricks to match a shape, aiming instead to create a stable configuration that complies with physical principles, reduces material usage, and maintains structural integrity under load. This problem introduces a new layer of complexity, as the validity of an action depends not only on local geometric constraints (e.g., element non-intersection) but also on its global impact on the entire structure's stress distribution.

The key insight driving our work is that effective structural construction requires reasoning at multiple levels of abstraction: strategic planning for overall structural layout and detailed execution for precise element placement. This naturally suggests a hierarchical approach in which different agents operate on different scales and optimize for different objectives. Our high-level planner focuses on strategic decisions about load distribution and structural stability, while our low-level executor ensures that individual brick placements satisfy local constraints.

Major Contributions:

- Novel Problem Formulation: We introduce stress minimization as a primary objective for 3D construction, moving beyond shape reconstruction to consider structural integrity and resource efficiency.

- Hierarchical Agent Architecture: We present a two-level system that integrates PPO-based strategic planning with WFC-based constraint satisfaction, enabling both global optimization and local constraint enforcement.
- Physics-Integrated Rewards: Our approach incorporates realistic stress analysis using finite element methods, providing physics-based feedback for structural optimization.
- Efficient Constraint Handling: Using WFC for low-level execution, we achieve efficient constraint satisfaction without the computational overhead of traditional planning approaches.

We validate our framework through a series of experiments that test its core components. The results first demonstrate the clear superiority of our hierarchical 3D planner over a simplified 2D version, confirming that full spatial reasoning is critical to achieving structural stability. We then show that the 3D planner's high performance is maintained when a fast neural operator replaces the traditional physics solver, establishing the viability of our approach for scalable, structurally-aware construction.

2 Related Work

2.1 Reinforcement Learning for Construction

Recent advances in RL-based construction have focused primarily on LEGO brick assembly tasks. Chung et al. introduced Brick-by-Brick, a pioneering approach that uses graph neural networks and action validity prediction for sequential brick placement, optimizing for IoU with target shapes (1). Their work demonstrated that incomplete target information (2D images) could be sufficient for constructing complex 3D objects through a comprehensive understanding of partial information and long-term planning, but was limited to shape reconstruction objectives. Budget-aware construction was explored by Ahn et al., who proposed BrECS, combining U-shaped convolutional networks with efficient constraint satisfaction for sequential brick assembly (3). Although their approach showed improvements in assembly speed and constraint handling, it remained focused on shape completion rather than structural optimization. More recently, multimodal approaches have emerged with CADrille, which combines point clouds, images, and text for CAD reconstruction (2). However, these approaches primarily target design reconstruction rather than structural construction with physical constraints.

Dillenburger presented a framework (13) that leverages PPO algorithms to optimize space layouts, showing how RL can accommodate offline tasks and seamlessly integrate with existing computer-aided design software.

2.2 Wave Function Collapse in Procedural Generation

The Wave Function Collapse (WFC) algorithm has emerged as a powerful tool for procedural content generation, particularly in scenarios requiring adherence to local constraint rules. Originally developed by Maxim Gumin for tile-based image generation, WFC operates by maintaining a superposition of possible states for each cell in a grid and progressively collapsing these possibilities based on local compatibility constraints (4). The effectiveness of the algorithm arises from its ability to generate coherent global patterns from simple local rules. Boris the Brave provides a comprehensive explanation of WFC as a constraint programming approach, where the computer uses built-in algorithms to find solutions to rigorously defined problems rather than following explicit imperative instructions (5). Recent developments in WFC have extended its applicability to 3D construction scenarios and multiscale generation problems. The integration of WFC with other procedural generation techniques has shown particular promise in creating complex, rule-constrained structures while maintaining computational efficiency.

2.3 Physics-Informed Neural Networks and Stress Prediction

The integration of physics-based constraints into neural network architectures has revolutionized computational mechanics and structural analysis. Physics-Informed Neural Networks (PINNs) have demonstrated remarkable success in solving partial differential equations while incorporating domain knowledge directly into the learning process (6; 7; 8). Fourier Neural Operators (FNOs) represent a significant advancement in this field, enabling the learning of operators that map between infinite-dimensional function spaces. Li et al. demonstrated that FNOs can efficiently capture complex operators arising in highly nonlinear PDEs, particularly those with high-frequency modes. The application of FNOs to the prediction of the stress-strain field in composite materials has shown exceptional promise (9; 10; 11).

3 Problem formulation

3.1 Construction Environment

We define our construction environment as a bounded 3D space of size $L \times W \times H$ discretized into unit cells. The construction process involves the sequential placement of LEGO-style bricks with fixed dimensions (l_b, w_b, h_b) in our base configuration. Each placement must satisfy three fundamental constraints:

1. **Support Constraint**: Bricks at height $z > 0$ must be supported by at least one existing brick

2. **Non-overlap Constraint**: New bricks cannot occupy space already filled by existing bricks

3. **Boundary Constraint**: All bricks must lie entirely within the construction space.

The state is represented as a binary occupancy grid of size $L \times W \times H$, where each cell contains a value of 1 if it is occupied by a brick and 0 otherwise. Due to the hierarchical nature of our approach, the action spaces for the high- and low-level agents are distinct and are described in detail in Sections 4.1 and 4.2, respectively.

3.2 Stress-Based Objectives

Unlike previous work that optimizes for shape similarity, we formulate construction as a multiobjective optimization problem.

The primary objective is to identify a valid structure s that minimizes von Mises stress:

$$\min_{s \in valid} \max_{detail \in s} \sigma_{vm}(s, d)$$

Secondary Objectives:

- Height constraint: Achieve target height H_{target}

- Resource Efficiency: Minimize total number of bricks used

- Structural Completeness: ensure that the structure forms a connected component

Structural completeness is enforced by the use of action masking. Other objectives are optimized due to maximization of the combined reward function:

$$R_\sigma = \begin{cases} 1, & \text{if } \big(\sigma < \sigma_{\text{thld}}\big) \land \big(h = h_{\text{target}}\big) \\ 0, & h < h_{target} \\ 0.5 \cdot \exp\left(\dfrac{\sigma_{\text{thld}} - \sigma}{g \cdot \sigma_{\text{thld}}}\right) & \text{otherwise} \end{cases} \tag{1}$$

$$R_h = w_h \cdot \max\left(0, \frac{h_{\text{target}} - |h_{\text{target}} - h|}{h_{\text{target}}}\right) \tag{2}$$

$$R_{\text{vol}} = \begin{cases} w_{\text{vol}} \cdot \dfrac{V_{\max} - N_{\text{cubic}}}{V_{\max} - h_{\text{target}}}, & \text{if } \big(\sigma < \sigma_{\text{thld}}\big) \land \big(h = h_{\text{target}}\big) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$R = \begin{cases} 1 + R_{\text{vol}}, & \text{if } \big(\sigma < \sigma_{\text{thld}}\big) \land \big(h = h_{\text{target}}\big) \\ R_\sigma + R_h, & \text{otherwise} \end{cases} \tag{4}$$

where σ is the maximum von Mises stress in the structure and σ_{thld} is the material-dependent stress limit.

The design of the reward function is crucial for guiding the agent toward desirable solutions. Our parameter choices are guided by a two-stage learning strategy: first, achieve structural viability, and second, optimize for resource efficiency.

- Stress threshold σ_{thld}: this parameter defines the boundary for a structurally sound design. The threshold is derived from the properties of the brick material. Any structure whose maximum von Mises stress exceeds this value is considered a failure.
- Height and stress reward (w_h, R_σ): the primary goal is to train an agent that can successfully build a structure of the target height that also meets the stress constraint. The reward for any "unsuccessful" state (either $h < h_{target}$ or $\sigma >= \sigma_{thld}$) is structured to be strictly less than the reward for a "successful" state. A successful construction receives a base reward of 1, plus a volume bonus. We set $w_h = 0.3$. This ensures that even if an agent builds to the full target height (earning the maximum $R_h = 0.3$), but fails the stress test (earning a maximum possible R_σ of 0.5), the total reward ($R = R_\sigma + R_h \approx 0.8$) remains definitively below the success reward of 1.0. This gradient strongly incentivizes the agent to first satisfy the hard constraints.
- Volume reward (w_{vol}, R_{vol}): the R_{vol} term acts as a bonus reward, applied only after a successful construction is achieved. It introduces the trade-off between the resource economy and structural robustness. A higher w_{vol} encourages the agent to use the fewest bricks possible, potentially creating solutions that are closer to the stress threshold. A lower w_{vol} allows the agent to build more robust, over-engineered structures.

4 Proposed Method

4.1 High-Level Planner

Operates on a coarse grid (e.g. $10 \times 10 \times h_{target}$) and makes strategic decisions. After the planning is completed, the high-level plan is stretched to the size of the scene.

Action Space:

- Terminal action: Binary choice to stop building.
- Position Selection: $L \times W$ possible positions on a 2D grid.

- Shape Selection: Multiple rectangular shape configurations (a rectangular shape consists of multiple bricks).

Responsibilities:

- Plans the overall structure layout and makes strategic decisions about placement of large shape components.
- Optimizes structural stability and stress distribution using physics-based reward signals

Network Architecture. Our architecture combines:

- A image encoder which consists of Conv3d layers
- Action masking for enforcing construction constraints

The extracted features are flattened and passed through fully connected 'actor' layers that output an embedding, which then feeds into separate action heads for selecting block colors, shapes, and placement coordinates. This hierarchical processing enables spatial awareness and specialized decision-making for each action component.

Training Flow. The training algorithm is Proximal Policy Optimization (PPO).

1. The high-level agent (PPO) plans each layer (z-level).
2. The low-level agent (WFC) completes the detailed implementation.
3. Physics computation using FEA or Neural Operator.
4. Reward calculation based on structural stability.

4.2 Low-Level WFC Executor

The low-level execution is performed using a modified Wave Function Collapse (WFC) algorithm specialized for brick assembly. Operates on a fine grid (e.g. $20 \times 20 \times h_{target}$), which is divided into frames of fixed size $F \times F$.

Action Space:

- Discrete actions: $F \times F \times 2 - 1$ possible actions
- Each action represents position (x, y) and orientation (0 or 1)

Responsibilities:

- Executes the detailed placement of bricks
- Optimizes for IoU (Intersection over Union) with target plan
- Handles fine-grained structural details

IoU Optimization. Within WFC constraints, the chosen brick placement maximizes the Intersection-over-Union (IoU) with the high-level plan:

$$\text{IoU} = \frac{|\text{Plan} \cap \text{Current}|}{|\text{Plan} \cup \text{Current}|}$$

This ensures that local executions adhere to both constraint satisfaction and alignment with high-level design intent.

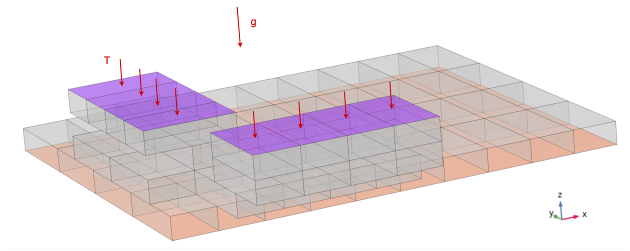


Figure 1: Boundary conditions

4.3 Physics Integration

4.3.1 FEA

Problem Formulation and Boundary Conditions. To evaluate structural stability, we develop a comprehensive finite element model that simulates realistic loading scenarios for brick assemblies. The model incorporates two primary loading mechanisms: gravitational body forces and external surface loads. The upper surface of the constructed structure experiences a uniformly distributed load, simulating potential operational loads, while the bottom surface is fully constrained with fixed boundary conditions to represent foundation support. Gravitational acceleration $\left(g = 9.81 \text{ m/s}^2\right)$ is applied as a body force throughout the structure volume.

The boundary conditions are:

- fixed displacement constraints at the base (u = 0),
- free boundaries on the sides, and prescribed surface traction on the top surface.

Mathematical Formulation. The structural analysis is governed by the linear elasticity equations under static equilibrium conditions:

$$\begin{aligned}
&\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} = 0 \\
&\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\epsilon} \\
&\boldsymbol{\epsilon} = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)
\end{aligned}$$

The von Mises stress is given by:

$$\sigma_{\text{von Mises}} = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}{2}}$$

where $\boldsymbol{\sigma}$ represents the stress tensor, ρ is material density, $\mathbb{C}$ is the fourth-order elasticity tensor, $\boldsymbol{\epsilon}$ denotes the strain tensor, and $\mathbf{u}$ is the displacement field. The constitutive relationship assumes isotropic linear elastic behavior with material properties.

Numerical Implementation. During training, stress analyses were performed asynchronously via a custom cloud service employing DOLFINx (FEniCS project) for finite element analysis. This asynchronous approach enabled parallel processing of multiple structural configurations, significantly reducing computational overhead during reinforcement learning training.

For additional validation and benchmarking, selected configurations were also analyzed using COMSOL Multiphysics to verify consistency of predictions across different solvers. Mesh resolution was chosen at a moderate level, providing a balance between computational efficiency and accuracy. This approach was sufficient for the requirements of current experiments, where rapid feedback and scalability were prioritized over high-fidelity resolution.

4.3.2 Neural Operator

In the reinforcement learning framework for the 3D lego brick assembly, a major computational bottleneck arises from the physics simulation required to evaluate the von Mises stress for the reward calculation. Traditional finite element analysis (FEA) solvers, despite their accuracy, are computationally expensive and significantly limit the sample efficiency of RL training. To overcome this, we implement a physics approximator based on Fourier Neural Operators (FNOs), which provides rapid and accurate stress field predictions from geometric data, enabling efficient and scalable training.

Architecture and Implementation Our physics approximator leverages a 3D Fourier Neural Operator architecture tailored to predict stress fields from signed distance function (SDF) representations of the built structures. The key details of the model consist of the following:

- Input Processing: The model receives single channel SDF input with spatial dimensions [64, 64, 12], coding the assembled brick configurations in 3D space.
- FNO Core: A 4-layer neural network with 16 Fourier modes per spatial dimension and 32 hidden channels, using a projection channel ratio of 4 to balance accuracy and computational cost.
- Output generation: The network outputs single-channel stress field maps representing the von Mises stress distribution throughout the structure.
- Training setup: Training is carried out over 700 epochs using the AdamW optimizer (learning rate 3e-4, weight decay 1e-3) with a cosine annealing learning rate scheduler to promote stable convergence.

Integration with RL Environment The trained FNO model replaces the traditional FEA solver in the RL environment's reward pipeline:

1. Convert the current assembly state into its SDF representation.
2. Use the FNO to rapidly predict the full 3D von-Mises stress field in milliseconds.
3. Extract stress metrics from the prediction to calculate the reward for structural stability.

This approach represents a pivotal advancement for integrating realistic physics into reinforcement learning for structural assembly, where computational efficiency is crucial without compromising fidelity. It allows the RL agent to learn to build stable, minimal-stress 3D structures from bricks effectively and efficiently.

5 Experiments

The implementation utilizes JAX/Flax for efficient GPU acceleration and automatic differentiation.

5.1 Experimental Setup

We implemented an environment for 3D structural design optimization, where agents must assemble stable structures within physical constraints. The setup features two agent levels: a high-level agent (feudal) responsible for strategic spatial planning, and low-level agents (vassal) performing detailed block placement according to the high-level plan.

Environment Configuration

- **Grids:** High-level agent operates on a $10 \times 10 \times 4$ grid. Low-level agents act within $20 \times 20 \times 4$ local regions.
- **Target height:** 4 units.

- **Maximum blocks:** 72 per structure.
- **Episode length:** Up to 25 steps.
- **Parallelization:** 20 concurrent training environments for improved sample efficiency.

Physics Simulation

- **Structural analysis:** Realistic mechanics via finite element analysis (FEA), using a Dolfinx-based solver.
- **Stress threshold:** 4×10^8 Pa (configurable).
- **Material:** Aluminum (Young's modulus $E = 68$ GPa, Poisson's ratio $\nu = 0.30$, density $\rho = 2698.9$ kg/m^3).
- **Applied force:** 3×10^6 N uniformly distributed.
- **Mesh parameters:** Factor 0.7.

Reward Structure Multi-objective reward balances feasibility, efficiency, and quality:

- **Stress-based reward:** 1.0 for stress below threshold and full height, shaped as $1 - 0.5 \times (\text{stress/threshold})^2$ if stress is below threshold. Penalty -1.0 for unrealistically low stress (< 1000 Pa).
- **Material efficiency:** Weighted by $w_{\text{volume}} = 0.5$, incentivizing lower block counts.
- **Height completion:** Weighted by $w_{\text{height}} = 0.5$, encouraging target achievement.

Action Space The high-level agent samples from a continuous action space:

- **Shape vector:** Encodes geometry.
- **Position vector:** (x, y, z) placement.
- **Color vector:** For block-type encoding (visualization only).

State Representation Observations include:

- **3D occupancy grids:** $10 \times 10 \times 4$ (feudal), $20 \times 20 \times 4$ (vassal).
- **Stress maps:** the maximum value of the calculated stress for each cube.
- **History:** Construction steps and current height.
- **Progress** and completion flags.

This setup enables systematic investigation of how hierarchical learning architectures can solve physically constrained engineering problems, requiring an interplay between high-level strategy and low-level execution.

Baseline Comparisons To assess the effectiveness of our construction method with stress-aware learning, we compare it to a simplified baseline based on a 2D high-level planner. This baseline restricts high-level decisions to two spatial dimensions: at each height layer, it plans the placement of components without reasoning about full 3D connectivity between layers. The rest of the pipeline—including low-level brick placement using WFC, constraint enforcement, and physics-based reward functions—remains identical. This controlled comparison directly isolates the benefits of full spatial (3D) reasoning for structural stability, efficient material usage, and stress minimization.

5.2 Neural Operator training

The FNO approximator delivers the following:

- Speedup: 85x to 150x faster stress evaluations compared to conventional FEA.
- Scalability: Efficient use of memory and computation for 3D geometric input, suitable for large-scale problems.

Training data consists of paired SDF and FEA-derived stress maps from a diverse set of brick assembly configurations:

- Geometric Diversity: The structural arrangements range from 20 to 200 bricks in various spatial layouts.
- Stress Classification: Assemblies are designated as "good" if their stress is below $4 \cdot 10^8$ Pa, whereas those exceeding this limit are considered as "bad," guiding the learning toward safe designs.
- Stratified Sampling: Guarantees a balanced representation across different brick counts and stress categories.
- Data Preprocessing: Stress values are logarithmically scaled to stabilize training.

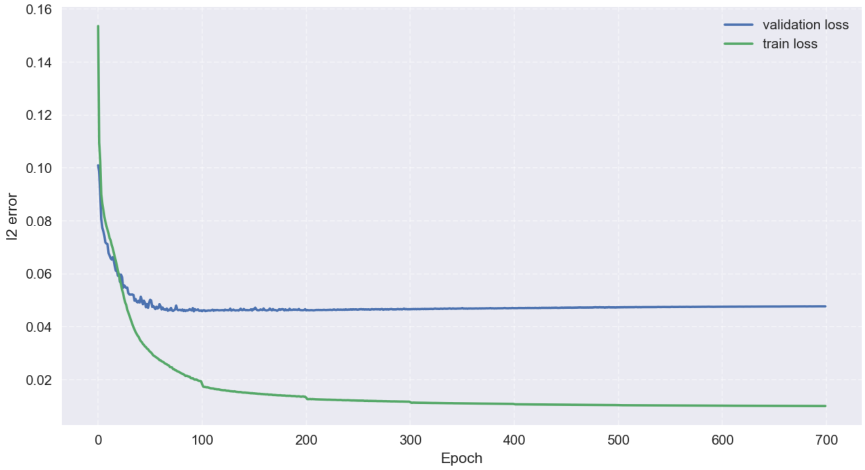


Figure 2: Convergence of the Fourier Neural Operator for Stress Prediction, shown from epoch 1.

The training performance of the Fourier Neural Operator is illustrated in Figure 2. The plot omits the initial value to prevent a large drop in error (from ~ 1.5 to ~ 0.15) from compressing the y-axis and causing the subsequent convergence to appear negligible. The model validation loss converges rapidly, achieving a minimum of approximately 5% l2-error within the first 80 training epochs. This fast convergence leads to a model that accurately estimates the stress fields in a diverse validation set, which was constructed using stratified sampling to ensure a rich mix of geometric configurations and stress profiles.

Qualitatively, the predicted and true values are very similar: both exhibit tiny stress values in the first and second stages, with maximum stresses concentrated on the upper floor (see Figures 3 and 4). The only notable difference is that the true high-stress values form two distinct clusters, whereas the neural operator predicts a larger contiguous high-stress area that includes both clusters, as well as some space between and around them. Quantitatively, the values are also close: the maximum predicted stress is 3.2×10^9, while the maximum true stress is 3.5×10^9. This corresponds to a relative error of 8.6% in predicting maximum stress, which is acceptable for the calculation of rewards in the reinforcement learning algorithm.

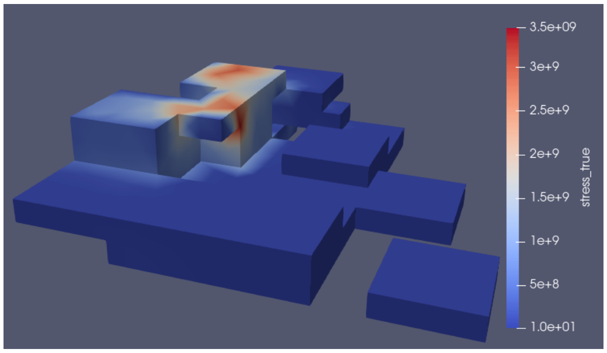


Figure 3: Von Mises stress values computed by the FEM solver

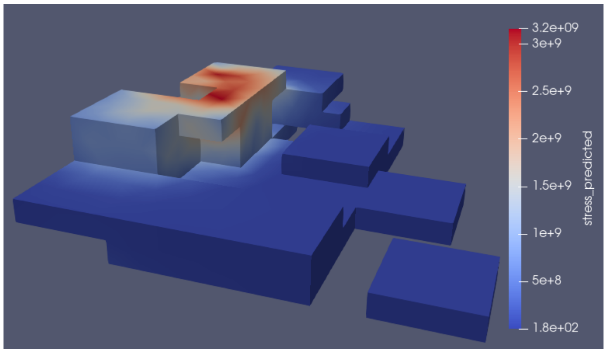


Figure 4: Predicted von Mises stress values by the neural operator

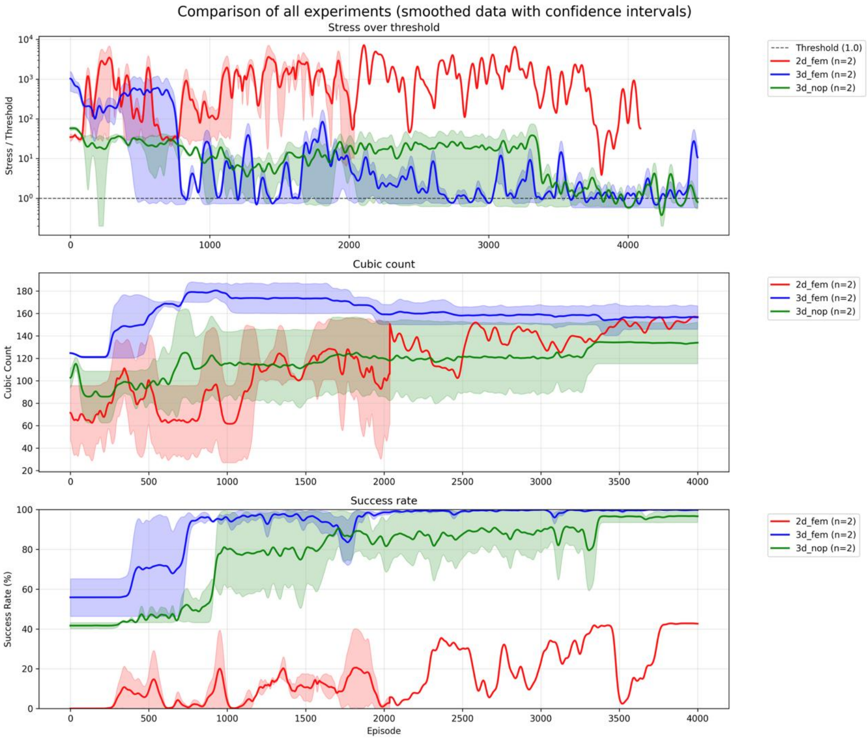


Figure 5: Performance comparison of planner configurations. The plot displays training metrics for three agents: a 2D planner using FEA (red), a 3D planner with FEA (blue), and a 3D planner utilizing the Fourier Neural Operator (green).

5.3 Structural Performance Results

Figure 5 presents the main quantitative results of our experiments, comparing the performance of our proposed 3D planner (with the FEA and the FNO approximator) against the 2D planner baseline in three key metrics: structural stability, resource efficiency, and overall construction success. The solid lines represent the mean performance averaged in three independent runs with different random seeds. The shaded areas show the corresponding minimum and maximum values of those runs, illustrating the performance variance.

Structural Stability (Stress over Threshold): The primary indicator of a successful design is the stress-to-threshold ratio, shown in the top plot. A value below 1.0 signifies that the maximum von Mises stress is within the acceptable material limits. As illustrated, both 3D planners successfully learn policies that reduce the stress-to-threshold ratio below 1.0, whereas the 2D planner consistently fails to do so. A closer comparison of the 3D planners reveals that the agent guided by the FEA solver exhibits faster convergence, achieving acceptable performance in fewer than 3,000 PPO iterations. However, the FNO-based agent also reliably converges to the required performance level.

Overall Performance (Success Rate): The bottom plot provides the percentage of episodes in a batch that result in a complete structure (that is, the target height is met and the stress-to-threshold ratio is below 1.0). This metric offers a decisive look at overall performance. Both 3D planners (blue and green) converge to near-perfect success rates, with the operator-based agent requiring more interactions to do so. In contrast, the 2D planner exhibits strong fluctuations and peaks at only about 40%, underscoring its instability and limited effectiveness.

Resource Efficiency (Cubic Count): The middle plot shows the number of bricks used by each agent. This metric reveals the trade-off between the resource economy and structural integrity. Although using fewer bricks is efficient, building a stable, full-height structure is difficult with too few materials. In contrast, a high brick count does not guarantee low stress.

The plot shows that all agents initially increase the brick count to expand the load-bearing area at the top layer, which lowers the stress-to-threshold ratio by distributing loads more evenly. The 2D planner demonstrates an upward trend across training, which is consistent with its low success rate: because unsuccessful episodes do not receive the volume bonus, there is little incentive to reduce brick usage, while the stress-driven component still rewards reductions in peak stress—often achieved by adding bricks. By contrast, the 3D planner's brick count (blue) begins to decline after roughly 1,000 iterations, coinciding with near-perfect success rates; this situation indicates that the volume bonus starts to dominate the optimization, guiding the agent to reduce material while maintaining structural feasibility.

Taken together, these results highlight the clear superiority of the hierarchical 3D planning approach. Furthermore, they provide strong evidence that Fourier Neural Operator can serve as a high-fidelity, computationally efficient substitute for traditional FEA solvers within the reinforcement learning loop, enabling effective and scalable training for complex, physics-constrained tasks.

State Space Complexity Analysis: The state space in our construction environment scales exponentially with grid resolution. For a $20 \times 20 \times 4$ voxel grid with binary occupancy states, the theoretical state space encompasses $2^{1600} \approx 10^{482}$ possible configurations. When constrained to valid $4 \times 2 \times 1$ brick placements, each brick admits 2,584 distinct placement positions, yielding combinatorial complexity that renders exhaustive search intractable.

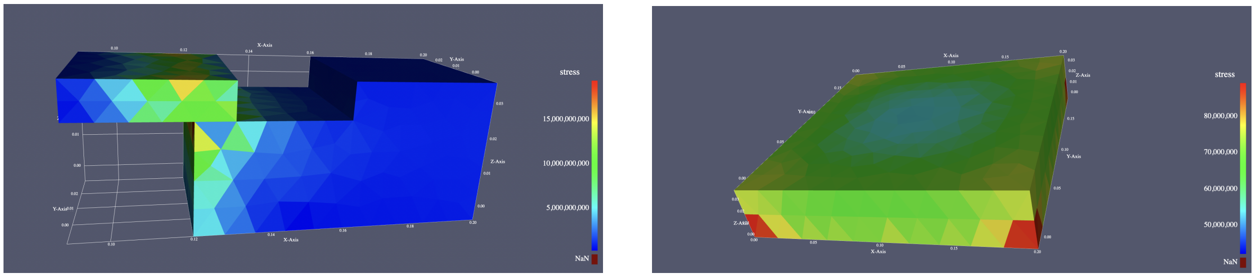
Our hierarchical agent architecture successfully identifies structurally sound, material-efficient solutions within this space. The key insight lies in the emergence of implicit construction heuristics not explicitly programmed into the reward function:

- **Structural coherence**: Avoiding unsupported elements without explicit geometric constraints
- **Load distribution**: Utilizing symmetrical configurations to minimize stress concentrations
- **Material efficiency**: Discovering designs that satisfy stability requirements with minimal resources

5.4 Qualitative Analysis

To further elucidate the learning dynamics and qualitative outcomes of our approach, we provide visual examples of structures generated at different stages of training, focusing on stress distributions and structural characteristics.

During the initial training phase, the model frequently generated suboptimal configurations (Figure 6a), including designs with hanging or poorly supported bricks. These structures were unable to meet the allowable stress criteria, as evidenced by substantial stress concentrations that exceeded the material threshold.



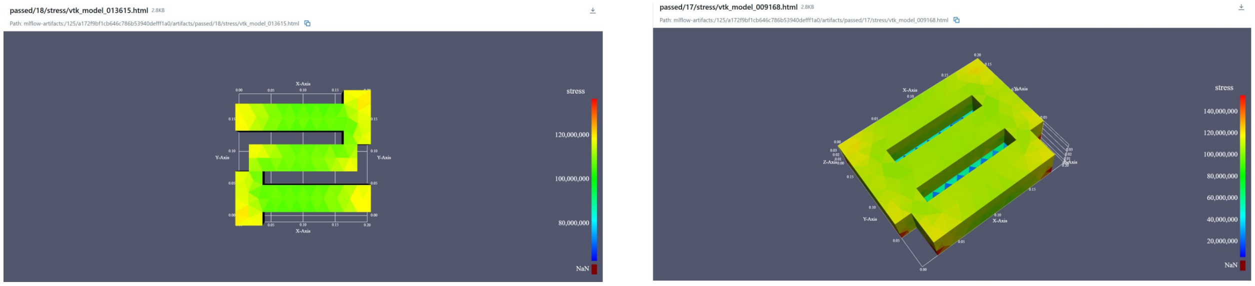
(a) Such designs often failed to satisfy the prescribed stress threshold and frequently contained overhanging elements with inadequate support.

(b) The stress is more evenly distributed, with the majority of the structure satisfying the threshold, though isolated high-stress regions remain.

Figure 6: Early and intermediate-stage structures generated during training. The left panel illustrates a failure case with high stress concentrations, while the right panel shows a partially stabilized structure where most stresses remain below the critical threshold, although some localized peaks persist.

As training progressed, the agent learned to construct more stable assemblies (Figure 6b). The frequency of failure cases decreased and the stress distribution became more uniform, although in certain regions local peaks in stress still appeared. These intermediate solutions often met the target height and connectivity requirements, but did not optimize material efficiency yet.

In the later stages of training, the model consistently produces high-quality structures (Figures 7a and 7b). These solutions not only conform to the target stress limits, but also significantly reduce the number of bricks required, demonstrating the efficiency of learned material without sacrificing structural integrity. The stress gradients are smoother and critical regions are reinforced, reflecting advanced planning and a refined balance between stability and resource usage.


passed/18/stress/vtk_model_013615.html 2.8KB

Path: mlflow-artifacts:/125/a172f9bf1cb646c786b53940defff1a0/artifacts/passed/18/stress/vtk_model_013615.html

passed/17/stress/vtk_model_009168.html 2.8KB

Path: mlflow-artifacts:/125/a172f9bf1cb646c786b53940defff1a0/artifacts/passed/17/stress/vtk_model_009168.html

(a) Mature design after extended training. The structure is compact, uses fewer bricks, and successfully maintains stresses below the prescribed limit. This demonstrates the agent's ability to discover both efficient and physically valid solutions.

(b) Alternative top-down view of an optimized design at convergence. The agent has learned to maximize spatial efficiency while ensuring all stress regions remain safely below the failure threshold.

Figure 7: Mature structures achieved after extended training. Both designs meet the stress constraints while using significantly fewer blocks. The solutions demonstrate the agent's ability to optimize both material efficiency and structural reliability.

In general, these results illustrate a key property of our approach: the ability to autonomously discover design strategies that satisfy physical requirements and optimize resource consumption. The qualitative progression from failure cases to threshold-compliant solutions to resource-effective constructions demonstrates that the proposed framework is capable of sophisticated, constraint-aware reasoning in complex structural assembly tasks.

6 Discussion

6.1 Key Insights

Hierarchical Decomposition Benefits: Our results demonstrate that separating strategic planning from tactical execution leads to improved structural performance. The high-level planner can focus on global optimization, while the low-level executor ensures local constraint satisfaction.

Despite the apparent simplicity of our discrete assembly task, the approach demonstrates effective navigation of a large solution space. The hierarchical learning architecture enables the agent to discover construction strategies and structural principles that extend beyond the explicitly encoded reward signals.

Physics-Based Learning: Incorporating stress analysis directly into the reward function enables the learning of structurally sound construction strategies that would be difficult to discover through shape-based objectives alone.

Constraint Satisfaction Efficiency: The WFC-based executor provides efficient constraint handling compared to traditional planning approaches, enabling real-time construction with complex constraint sets.

6.2 Limitations

Computational Overhead: Physics simulation introduces computational cost, though our asynchronous implementation mitigates this impact.

Simplified Physics: Our current implementation uses simplified material models and loading conditions. Extension to more complex structural analysis remains for future work.

Limited Brick Variety: Current experiments focus on single brick type. Extension to multiple types of bricks and complex shapes is planned.

7 Conclusion

This study demonstrates that hierarchical decomposition, paired with physics-integrated rewards, enables agents to discover structures that satisfy stress constraints and minimize material, surpassing a 2D planning alternative that lacks full spatial reasoning. The low-level WFC executor complements the high-level policy by enforcing local feasibility and alignment with the global plan, supporting consistent success across runs and seeds. Moreover, an FNO-based surrogate effectively replaces FEM for reward computation without degrading ultimate performance, offering a practical path to scalable training in physics-constrained design tasks.

References

[1] H. Chung, J. Kim, B. Knyazev, J. Lee, G. W. Taylor, J. Park, and M. Cho. Brick-by-brick: Combinatorial construction with deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021. proceedings.neurips.cc.

[2] M. Kolodiazhnyi et al. cadrille: Multi-modal cad reconstruction with online reinforcement learning. *arXiv preprint arXiv:2505.22914*, 2025. arXiv:2505.22914.

[3] S. Ahn, J. Kim, M. Cho, and J. Park. Budget-aware sequential brick assembly with efficient constraint satisfaction. *Transactions on Machine Learning Research*, 2024. openreview.net.

[4] M. Gumin. Wavefunctioncollapse. GitHub repository, 2016. github.com/mxgmn/WaveFunctionCollapse.

[5] B. The Brave. Wave function collapse tips and tricks. 2020. boristhebrave.com.

[6] A. Kashefi and T. Mukerji. A general method for solving differential equations of motion using physics-informed neural networks. *arXiv preprint arXiv:2211.16190*, 2022. arXiv:2211.16190.

[7] H. Bolandi, M. A. El-Gohary, S. K. Gad, and A. M. Saleh. Physics informed neural network for dynamic stress prediction in porous media. *Artificial Intelligence Review*, 56(1):1–25, 2023. doi.org/10.1007/s10489-023-04923-8.

[8] S. Lee, J. H. Kim, J. W. Kim, and H. J. Yoon. Predictive stress analysis in simplified spinal disc model using physics-informed neural networks. *Computer Methods in Biomechanics and Biomedical Engineering*, pages 1–13, 2025. doi.org/10.1080/10255842.2025.2471504.

[9] M. M. Rashid, T. Pittie, S. Chakraborty, and N. M. Krishnan. Learning the stress-strain fields in digital composites using fourier neural operator. *arXiv preprint arXiv:2207.03239*, 2022. arXiv:2207.03239.

[10] M. S. Khorrami, C. C. Roth, J. A. Tempelman, F. Marefat, D. S. Dikbas, B. Gault, S. Zaefferer, and D. Raabe. A physics-encoded fourier neural operator approach for surrogate modeling of divergence-free stress fields in solids. *arXiv preprint arXiv:2408.15408*, 2024. arXiv:2408.15408.

[11] Y. Shin, J. Song, S. Lee, M. Ko, and S. Jeon. Local stress field prediction from global displacement using fourier neural operator. *Journal of Computational Design and Engineering*, 12(5):21–34, 2022. doi.org/10.1093/jcde/qwac112.

[12] G. Kour and R. Saabne. Real-time segmentation of on-line handwritten arabic script. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 417–422. IEEE, 2014.

[13] R. Kakooee and B. Dillenburger. Reimagining space layout design through deep reinforcement learning. *Journal of Computational Design and Engineering*, 11(3):43–55, 2024. Oxford University Press.

A Implementation details

A.1 Image Encoder

The high-level PPO planner employs a 3D convolutional neural network (CNN) encoder to process the 3D grid-based environment. The encoder consists of a two-layer 3D CNN with ReLU activations: the first convolution layer maps from 1 input channel (binary occupancy) to 32 channels using a 3×3×3 kernel, followed by a second layer transforming 32 channels to 16 output channels with the same kernel size. Padding and stride settings maintain the spatial resolution throughout the encoding. The encoder thus preserves spatial structure and encodes volumetric features relevant for planning.

A.2 Hyperparameters

Parameter	Value
seed	42
coarse grid size	10
fine grid size	20
target height	4
num_envs	20
feudal use augmentation	True
wfc greedy collapse	True

Table 1: Main hyper-parameters

Parameter	**Value**
embedding	standard
num episodes	1,000,000
gamma	0.999
conv output channel	16
activation function	tanh

Table 2: Hyper-parameters from `3d_feudal.yaml`.

Parameter	**Value**
total_timesteps	1,000,000
buffer_size	1
num_episodes_in_batch	25
anneal_lr	True
update_epochs	3
tau	0.95
norm_adv	True
clip_coef	0.2
clip_vloss	True
entropy_coeff	0.0015
critic_coeff	0.5
gradient_clip	5.0
target_kl	0.005
num_minibatches	1
learning_rate	0.0003
gamma	0.99
separate_aug	False

Table 3: PPO hyper-parameters from `ppo.yaml`.