Zhong, Ximing; Koh, Immanuel; Fricker, Prof. Dr. Pia

# Building-GNN: Exploring a co-design framework for generating controllable 3D building prototypes by graph and recurrent neural networks

*Published in:*
Digital Design Reconsidered

*DOI:*
10.52842/conf.ecaade.2023.2.431

Published: 01/01/2023

*Document Version*
Publisher's PDF, also known as Version of record

*Please cite the original version:*
Zhong, X., Koh, I., & Fricker, P. D. P. (2023). Building-GNN: Exploring a co-design framework for generating controllable 3D building prototypes by graph and recurrent neural networks. In W. Dokonal, U. Hirschberg, & G. Wurzer (Eds.), *Digital Design Reconsidered: Proceedings of the 41st Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe 2023)* (Vol. 2, pp. 431-440). (eCAADe proceedings). eCAADe. https://doi.org/10.52842/conf.ecaade.2023.2.431

# Building-GNN

## Exploring a co-design framework for generating controllable 3D building prototypes by graph and recurrent neural networks

*Ximing Zhong[1], Immanuel Koh[2], Fricker Pia[3]*
*[1,3]Aalto University [2]Singapore University of Technology and Design*
*[1]ximing.zhong@aalto.fi*

*This paper discusses a novel deep learning (DL） framework named Building-GNN, which combines the Graph Neural Network (GNN) and the Recurrent neural network (RNN) to address the challenge of generating a controllable 3D voxel building model. The aim is to enable architects and AI to jointly explore the shape and internal spatial planning of 3D building models, forming a co-design paradigm. While the 3D results of previous DL methods, such as 3DGAN, are challenging to control in detail and meet the constraints and preferences of architects' inputs, Building-GNN allows for reasoning about the complex constraint relationships between each voxel. In Building-GNN, the GNN simulates and learns the graph structure relationship between 3D voxels, and the RNN captures the complex interplaying constraint relationships between voxels. The training set consists of 4000 rule-based generated 3D voxel models labeled with different degrees of masking. The quality of the 3D results is evaluated using metrics such as IoU, Fid, and constraint satisfaction. The results demonstrate that adding RNN enhances the accuracy of 3D model shape and voxel relationship prediction. Building-GNN can perform multi-step rational reasoning to complete the 3D model layout planning in different scenarios based on the architect's precise control and incomplete input.*

**Keywords:** *Deep Learning, Graph Neural Networks, 3D Building Layout, Co-design Recurrent Neural Networks, Multi-step Reasoning*

## INTRODUCTION

There is a growing interest in using deep learning (DL) to combine human creativity with machine intelligence and enhance the quality of the design process. Licklider (1960) proposed a powerful human-machine design scenario in which machine intelligence and human intelligence are connected through a fast network to explore new prototypes together. In the architecture field, many DL approaches can already collaborate with architects on 2D spatial layouts effectively using techniques such as Generative Adversarial Networks (GAN) and Variational Auto-Encoder (VAE) (Zhang & Huang, 2021). For example, the classical GAN networks that use Generator and Discriminator to play each other to simulate realistic 2D planar layout methods have been widely used in building planar design (Nauata et al., 2020). However, compared to the generation of 2D images, DL methods for 3D building shape exploration and spatial segmentation are still very challenging due to the higher dimensionality (Chang et al., 2021; Wu et al., 2016). The DL learning 3D voxelized models is widely used in architectural modeling (Chang et al., 2021). However, previous attempts, such as 3D-GAN are difficult to reason out complex constraint graph relationships between

each voxel unit, and the 3D results are difficult to control precisely (Li et al., 2019; Oussidi & Elhassouny, 2018; Zhong et al., 2022), as will be discussed in detail in later chapters. The generated 3D building may not conform to the constraints and preference input by the architect, resulting in significant modifications. These challenges highlight our research question of improving the control and accuracy of 3D building model generation while accommodating complex voxels constraints and architects' preferences.

We provide a framework called Building-GNN, in which GNNs are utilized to learn the intricate relationships between different voxels. RNNs are employed to perform multi-step reasoning to enhance the prediction accuracy of voxel layout by learning the interactions between node states and complex constraints in 3D models.

Our contribution is a new model for human-machine co-creation in 3D Building volumetric design that enables architects to input local constraints and explore the complete 3D building design morphology and internal spatial division directly and interactively in 3D space. Compared to previous 3D-GAN methods, our model allows architects to control each voxel of the 3D model precisely. And building-GNN can perform multi-step reasoning 3D model design options. The architect can rapidly explore a wide range of design possibilities while substantially ensuring that each design option adheres to the input constraints and preferences. The RNN parameters can control the degree of human and AI control capability on the final 3D result. Thus, humans and AI can cooperate to make decisions and control complex layouts without rigidly relying on predetermined programs (Licklider, 1960; Fricker et al., 2007). In summary, Building-GNN highlights the potential of GNN to improve the accuracy and control of generating 3D building models, suggesting a promising direction for addressing the challenges of 3D building shape exploration and spatial segmentation.

## GAN FOR 3D VOXEL MODEL GENERATION

Various approaches have been proposed to generate 3D voxel models using DL techniques. Some researchers have attempted to extract 2D features from 3D models and reconstruct 3D voxel models using encoding-decoding techniques. For instance, Zhong et al. (2022) generated 3D volumes by using GauGAN to learn the height information recorded in the graph color channel, while Zhang & Huang (2021) used styleGan to learn and generate continuous profile features, which were then reconstructed into 3D models. However, these encoding-decoding methods do not fully capture the 3D details of the building model, such as the detailed 3D model features in the middle of two consecutive profiles, which are ignored in Zhang's method (Zhang & Huang, 2021).

To completely learn the 3D model, Wu et al. (2016) proposed a 3D-GAN that can learn and generate a 3D voxel model similar to the dataset by sampling from a uniform noise distribution. Li et al. (2019) built a conditional GAN that can better guide the results of the 3D-GAN. Arshad & Beksi (2020) enhanced the structure of cGAN for predicting 3D point clouds with semantics. However, even with conditional GAN models, architects do not have complete interactive control over each voxel, and extensive manual modifications are still required to meet specific constraints and preferences. This results in a uni-directional, stepwise process, which does not allow AI and architects to jointly explore the final 3D model in a bidirectional co-design network, as proposed by Licklider (1960). To accurately extrapolate a complete 3D solution based on architects' voxel input, it is necessary to enhance the learning of complex graph structural relationships and constraints between individual voxel objects.

## GNN AND RNN FOR IMPROVING 3D BUILDING GENERATION.

GNNs are a class of DL networks used to process graph representation data with nodes and edges representing entities and their relationships. They can effectively aggregate information from adjacent

nodes in graph structures and predict node properties and complex relationships. In architectural design, GNNs are often applied to space layout problems based on graph representations. For example, Nauata et al. (2020) encoded the planar units and their adjacent relationships as graphs and then used GNN and GAN to generate 2D plan layouts. Chang et al. (2021) demonstrated the potential of GNNs to improve the control ability of 3D GANs by using GNNs to predict spatial node classification and then using these predictions as input conditions for a 3D GAN to generate 3D buildings that better match the architect's input. However, the final GAN result can deviate from the spatial node classification generated by GNNs. Additionally, while GNNs are good at learning the relationships between nodes and connected edges, they are inaccurate in reasoning the long-term dependencies between global nodes (Palm et al., 2018). In the discussed 3D building model design, the long-term dependencies between units mean that changes in each unit will affect and constrain the global network of all units, not just the direct connections.

Recurrent Neural Network (RNN) is a DL model suitable for processing sequential data to capture long-term dependencies in sequences. Palm et al. (2018) applied RNNs to reason about node relationships in graph-based relational networks and developed a high-accuracy DL model for complex constrained relationship reasoning tasks. Zhao et al. (2020) used GNN and GRU (a type of RNN) to predict traffic flows on street graph road networks at different time intervals and achieved high accuracy by capturing long-term dependencies with GRU. This evidence highlights the integration of RNNs into GNN frameworks has the potential to improve the accuracy of reasoning complex relationships between voxels in 3D models. This facilitates the development of a co-design network where architects can precisely modify each voxel, and AI can reason the global 3D voxel model, allowing them to work together to explore the final 3D model.

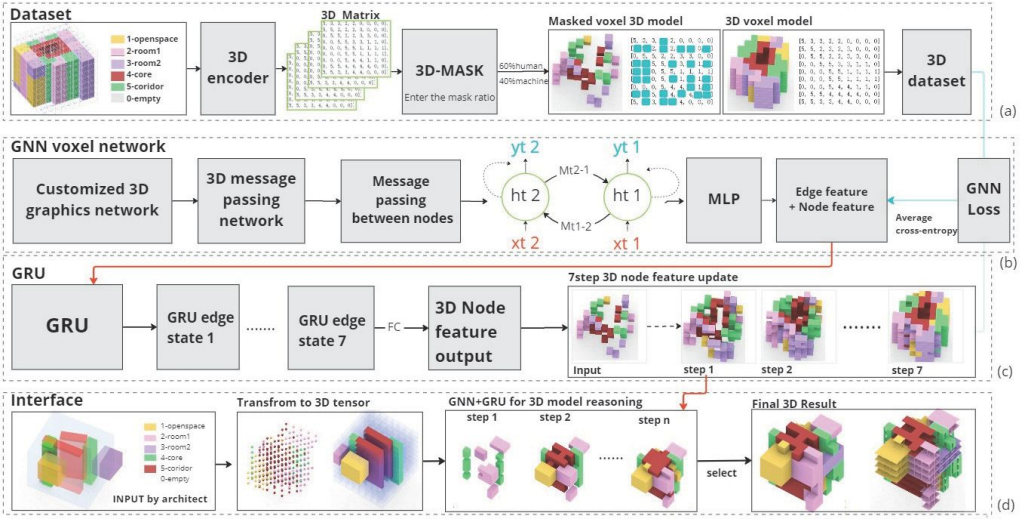


Figure 1
The co-design framework that leverages GNN+GRU to learn and predict 3D models

## FRAMEWORK

The building-GNN framework in this paper comprises three main components: data generation, the GNN and GRU network, and an interactive co-design module, as depicted in Figure 1.

**Research Objectives:** (1) To automatically generate various 3D building voxel models labeled with scaled voxel masks using a rule-driven approach. (2) To achieve high accuracy in predicting 3D shapes and internal voxel divisions in test sets using the GNN+GRU spatial network, with accuracy increasing as GRU iterations increase. (3) To enable interactive control of each spatial voxel while demonstrating good generalization capabilities to predict the boundaries of unlearned surface shapes. (4) To allow the GNN network to continuously move in the XYZ direction to expand the 3D voxel model to larger scales.

## Dataset generation

We employed an architect-parameterization approach inspired by Chang et al. (2021) to generate simplified voxel models of single-ring style office buildings, as shown in Figure 1(a). To introduce variety and plausibility to the dataset, we invited 15 architects to create multiple 3D model variants by modifying dimensional parameters and layout control lines. A movable sampler is used to sample 3D models at different angles and positions to improve the stitching of GNN-generated results. The 3D model variants and coding system are illustrated in Figure 2. The dataset consists of complete 3D models and partially masked models for GNN learning, with 4000 cases for training and 1000 for testing. We randomly masked 40% of the voxels and assumed that architect input of 60% content predicts 40% of the GNN results. The Results section will provide a detailed analysis of the size and masked proportion effects.

## Customized 3D graph network

The customized 3D graph network used in this study represents each voxel position $i \in [0,728]$ as a node in Figure 3(b), with the attribute $M_i$ indicating the semantic function of the voxel (ranging from 0 to 9, where 0 represents an empty voxel). Each node's neural network graph structure is denoted as $G_i$.

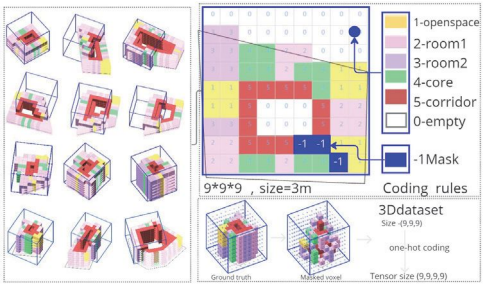


Figure 2
The dataset sampling and coding process.

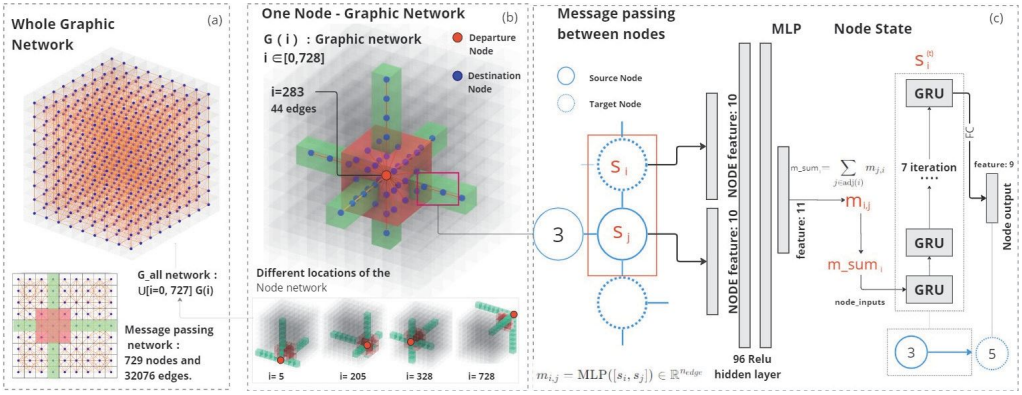


Figure 3
Graph structure and message passing network of Building-GNN.

To customize the voxel size of the GNN network based on the 3D model size requirements, we followed Palm et al. (2018 and used a 9*9 custom network that accurately predicts the constraints between units. For our experiments, we adopted the 9*9*9 voxel size, which is the same as the sampler size. As shown in Figure 3(b), each node in the graph structure is connected to its 333 neighboring voxel nodes, and all nodes in the XYZ direction of a given node are connected as neighbors. When a node is located on the network's edge or corner, it searches and connects to the closest 33*3 neighboring nodes inside the network, as illustrated in Figure 3(b). The message propagation network for the entire 3D network is defined as $U[0,729]G_i$, with 729 nodes and 32076 edges, as shown in Figure 3(a). This design enables messages to be propagated in all directions of XYZ, allowing for better learning of the constraints in XYZ direction and enabling the GNN network to fit larger scale models by moving in the XYZ direction.

## Combing GRU and message-passing network

The Message Passing Network (MPN) is constructed as follows:

1. The message input network consists of three fully connected layers. For each source-target node pair $(i, j)$, their state features are concatenated and input into an MLP network to obtain a message vector $m_{i,j}$. For each node $i$, the received messages from its neighbors are aggregated to obtain a message aggregation vector $m\_sum_i$, as shown in Figure 3 (c).
2. A GRU module is employed to update the state of each node $s_i^{(t)}$. The formula is as follows:
   $s_i^{(t)} = GRU\left([xi, m\_sum_i],\ \ s_i^{t-1}\right), \mathrm{t} \in [1, \mathrm{T}])$
   After each node $i$ receives its message aggregation vector $m\_sum_i$, it is concatenated with its current state feature vector$x_i$ , and the combination is fed into the GRU network along with the previous $s_i^{(t-1)}$. The states of each graph node are linearly combined to compute the output of the corresponding node at the current iteration. T denotes the number of state iterations.
3. The output network contains a fully connected layer whose output is a voxel's semantic classification by inputting $s_i^{(t)}$ .

**Training:** Adam optimizer with a learning rate of 0.001, branch=64. Number of iterations for GRU, T=6. The cross-entropy loss should be the average of the 6-step classification loss calculated at each iteration of GNN.

## Interactive 3D design interface

The interface is implemented using the GH-Python framework in the Rhino modeling platform. The architect can enter some constraints and preferred volumes, which are automatically converted into a numerical matrix. The building-GNN can reason the complete 3D voxel model in real-time based on the architect's input, as illustrated in Figure 1(d). The architect can then engage in a feedback loop with the AI, where the architect iterates on the results of building-GNN's reasoning to modify them according to their understanding of the site's design requirements.

## Evaluation of results

**Experiment 1:** We aim to test our model's reason capabilities for complex internal unit relationship layouts and 3D shapes. To do this, we randomly masked different levels of voxels on a Ground Truth (GT) parametric model and then compared the generated 3D model with the GT. We measured the accuracy using voxel accuracy (Vacc), Intersection-over-Union (IoU) score, and Frechet Inception Distance (FID) score, which considers the internal voxel semantics and ranges from 0 to 1. The IoU score is the primary evaluation metric, capturing the similarities and differences between the predicted and GT(Mo et al., 2019). The FID score is widely used to test for diversity and similarity (Nauata et al., 2020; Chang et al., 2021). The Vacc is used to help evaluate

the accuracy of the independent voxel between the generated model and GT.

**Experiment 2:** We tested the generalization ability of our model by generating 500 solutions with 15 professional architects to evaluate whether the GNN can generate reasonable 3D models for previously unseen design inputs. We also assess the architects' controllability by quantifying the extent to which the GNN satisfied boundary constraints for different spatial volumes entered by the architect. We calculated the Input Constraint Satisfaction (ICS) metric based on the IoU score between the architect's input and the GNN's output voxel models within various voxel semantic boundaries.

## RESULTS AND DISCUSSION

In this section, we evaluate the results of our building-GNN framework from various aspects and explore the model's shortcomings.

Quality of GNN-generated 3D voxel model results

The cross-entropy loss function converged to 0.27 after 16 iterations on a dataset of 4000 samples. The prediction accuracy improves with increasing iterations of the GRU update state, with different initial ratio voxel input conditions, each including 200 cases, as shown in Figure 4.

Compared to a single GNN model, our method accurately reasons the 3D model shape and predicts the internal voxel layout. As shown in Figure 5, our model can reasonably predict the complete 3D layout for different boundaries and input conditions, even for circular boundaries not included in the training set. However, the model's prediction accuracy decreased to 68.8% when faced with 20% fewer voxel inputs than shown in Figure 4. This is because our training set only included voxel inputs that are 60%, and the model did not learn cases with few voxel inputs. We need to adjust the mask ratio based on future 3D prediction needs.

To further evaluate our model, we randomly masked 20%-80% of the voxels in a thousand test models using a Gaussian distribution sampling method and used GNN to predict the solution. The FID, IoU, and Acc values for the test set are shown in Table 1. The FID values confirmed that the diversity of the GNN-generated 3D models decreased while the similarity increased, indicating that our model gradually learned the 3D shape and internal spatial division logic of the cases in the dataset. The final IoU accuracy of the model reached 0.883, indicating a high degree of accuracy in predicting the 3D voxel layout.

| | Vacc | FID | IoU |
|---|---|---|---|
| GNN | 0.6075 | 1.98 | 0.28 |
| GNN+GRU | | | |
| 1 | 0.75 | 0.623 | 0.678 |
| 2 | 0.781 | 0.151 | 0.810 |
| 3 | 0.831 | 0.182 | 0.826 |
| 4 | 0.837 | 0.161 | 0.839 |
| 5 | 0.835 | 0.083 | 0.857 |
| 6 | 0.852 | 0.072 | 0.883 |

Table 1
3D model result quality assessment of 1000 test set cases evaluated by ACC, FID, IoU

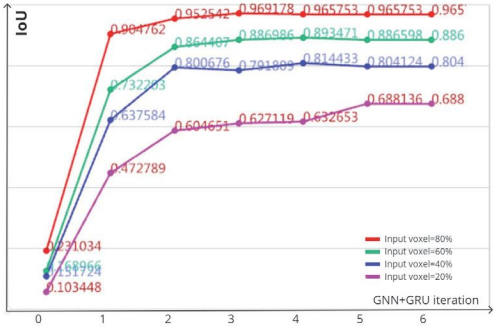


Figure 4
Comparison of IoU evaluation parameters for different scales of masked voxel input

## 3D result of the collaboration between architects and Building-GNN

The results presented in Figure 6 demonstrate that Building-GNN can effectively collaborate with architects and reason about 3D models in the face of different design input conditions, such as core cylinder count, architect input complexity, boundary shapes, and other factors. Our key findings include the following:

**Voxel layout relations in the results:** Even with local and ambiguous input volumes, building-GNN effectively identifies the spatial division of the voxel

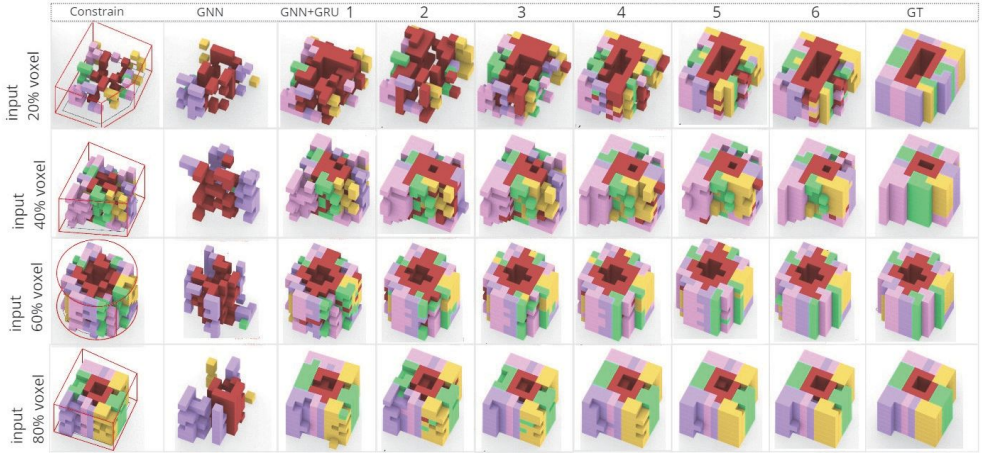


Figure 5
Comparison of the 3D model predictions of single GNN and our model with different GRU iterations

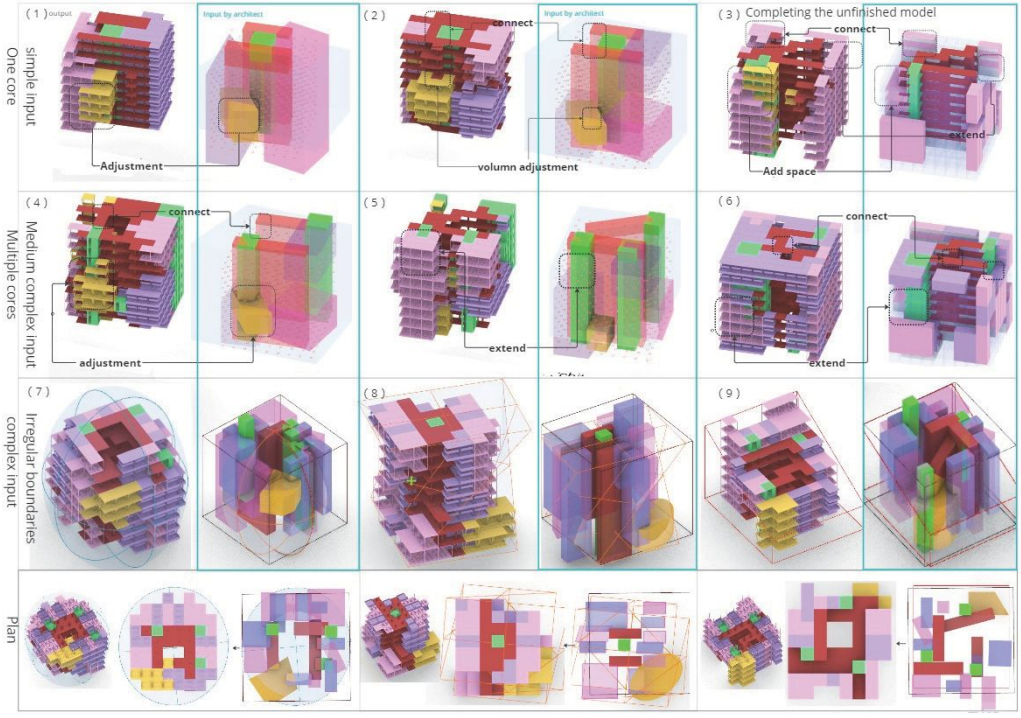


Figure 6
3D spatial divisions and shapes resulting from joint exploration by GNN and architects under varied design Input constraints and conditions.

model and it's 3D shape. The internal voxel partitioning in the results are similar to the training dataset when facing different complex input scenarios, such as those shown in Figure 6, including (2), (5), and (8). The floor plans in Figures 6 (7)-(9) demonstrate our method's ability to reason complex layout constraints between voxels and complete 3D models.

**Generalization Ability:** Despite our dataset consisting only of building models with simple polygonal single-loop corridor patterns, Building-GNN exhibits good generalization ability under more complex constraints and conditions. The model can generate 3D models subjected to specific design constraints, such as a spherical constraint shown in Figure 6 (7), and link multiple core cylinders in a trapezoidal building with horizontal corridors, as depicted in Figure 6 (9). The building-GNN can produce more diverse results with various design constraints than the training set.

**Traffic spaces:** Building-GNN generates traffic spaces that connect different volumes and make each room accessible, as exemplified in Figure 6 (2)-(8). The vertical traffic space can be customized by height while maintaining strict vertical alignment, as shown in Figure 6 (6). It can also automatically supplement horizontal corridors to form circular corridors, as illustrated in Figure 6(2).

**Spatial complements and modifications:** Building-GNN exhibits compliance with model boundaries in all scenarios and effectively learns the relationship between empty voxels and boundaries, as indicated in Figure 6 (8). Building-GNN can precisely fill in boundaries set by architects based on the surrounding spatial complements and extensions, as illustrated in Scheme (3) (6). Moreover, it can handle the articulation between different volumes to ensure accessibility. These results demonstrate how building-GNN excels at modifying the voxel relationship details between volumes input by architects to produce more reasonable layouts.

We quantify the architect's control ability over the final 3D results with different GRU parameters, as shown in Table 2. According to the ICS metrics in Table 2, on average, building-GNN's results comply with the architect's inputs about 94% in the final design. The architect maintains precise control over each voxel to explore with GNN for the final 3D model. The building-GNN modifies the horizontal traffic space of the architect's inputs the most, with ICS scores decreasing from 0.8 to 0.61, to ensure accessibility of the space, aligning with our observation of the results in Figure 6.

Building-GNN modifies some voxels of the architect's inputs to meet the complex voxel relationships it learned from the training set. As the GRU iterations increase, GNN modifies the architect's original inputs to a greater extent, resulting in smaller ICS scores. During the process of leveraging the strengths of architects and AI, the GRU iteration count enables us to quantitatively control the impact they have on the generation of 3D results.

| **GNN+GRU** | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| All volumes | 1.00 | 1.00 | 0.98 | 0.96 | 0.96 | 0.94 |
| Traffic space | 0.80 | 0.68 | 0.67 | 0.64 | 0.63 | 0.61 |
| Vertical Cores | 1.00 | 1.00 | 0.95 | 0.93 | 0.92 | 0.93 |
| Public space | 1.00 | 0.92 | 0.88 | 0.85 | 0.86 | 0.84 |
| Other room | 1.00 | 0.89 | 0.87 | 0.81 | 0.76 | 0.73 |

Table 2
The input constraint satisfaction (ICS) scores under different GRU parameters for diverse categories of spatial volumes input by architects.

## Results of stitching GNN networks

We experimented with the GNN sampler network in horizontal and vertical directions to stitch the 3D model, as illustrated in Figure 7 (a). The results show that GNN can complete the stitching to solve a larger range of architect input voxels without significant stitching traces. We quantify and compare the prediction results with the accuracy of GT using a model with a mask equal to 40% in the test set. The results show that as the size of the scale increases, the accuracy starts to decrease, as shown in Figure 7(b). This is probably because some GNN grids do

not have input voxel trails within them, thus producing erroneous judgments.

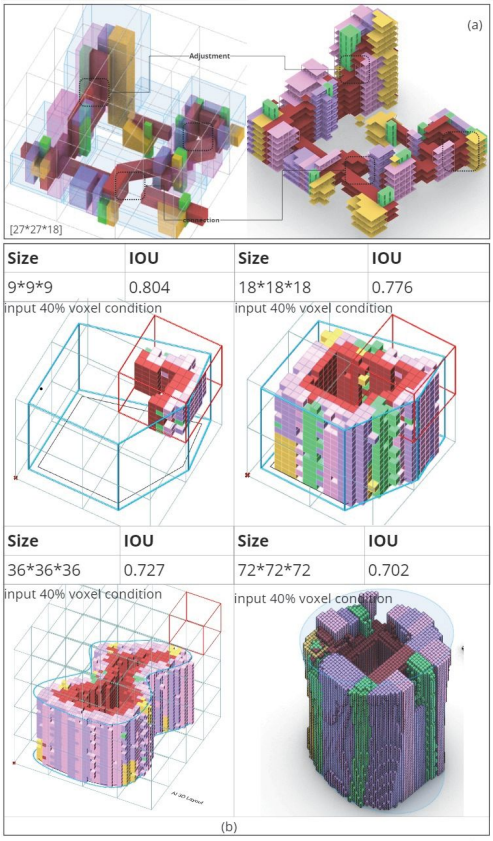


Figure 7
(a) Building-GNN splicing for 3D model prediction under expansive architectural input and site constraints. (b) Comparing prediction accuracy of Building-GNN network expansion across various scales of 3D model generation

## Limitations and future work

Our approach differs from previous 3D-GAN models in that it relies on the proportion of input voxel clues provided by the architect to predict accuracy due to the fixed 40% mask in our dataset. To address a variety of design scenarios, we plan to develop a systematic mask algorithm to train AI. Our framework can also integrate 3D-GAN to generate concept volumes that can be imported into building-GNN for voxel relationship inference and modification, providing architects with quick and creative options to complete 3D model designs. Although our method has demonstrated the effectiveness of our neural networks in handling complex 3D reasoning, training our models on large 3D datasets of different building types is still necessary to improve their performance in actual design projects. Furthermore, additional quantitative experiments are needed to determine the appropriate sample size, mask scale, and parameter settings to improve model accuracy. Additionally, our approach cannot predict wall positions for an exact building floor plan layout. To generate detailed 3D indoor layouts using GNN, we can integrate additional features into GNN nodes, such as materials, window numbers, internal wall layouts, and furniture information.

## CONCLUSION

Our research results demonstrate the efficacy of our co-design framework in accurately reasoning 3D voxel models while providing architects with control over the design process. This study makes a novel contribution to a human-machine framework in 3D architectural design, enabling architects to input local constraints and explore the entire 3D architectural shape and internal spatial partition directly in 3D space. It highlights the potential of GNN to enhance 3D generative models' accuracy and control in architecture. Our work reflects a promising direction to tackle challenges in 3D building shape exploration and spatial segmentation, allowing architects and AI to collaborate and flexibly create 3D models that meet complex site constraints and architect preferences within a 3D spatial network. Building-GNN serves as a partner that quickly completes detailed reasoning and provides specific prototypes to accommodate architect feedback. It not only utilizes AI to accelerate the design process but also leverages the architect's perception and understanding of the site, which can be difficult for a machine to surpass, to assist AI in interactively exploring 3D building layout results.

The feedback loop process between architects and AI has the potential to facilitate a closer, more creative, and flexible design process.

## REFERENCES


Arshad, M. S. and Beksi, W. J. (2020) 'A Progressive Conditional Generative Adversarial Network for Generating Dense and Colored 3D Point Clouds', in Proceedings of the 2020 International Conference on 3D Vision (3DV), IEEE, pp. 712-722.

Chang, K.-H., Cheng, C.-Y., Luo, J., Murata, S., Nourbakhsh, M. and Tsuji, Y. (2021) 'Building-GAN: Graph-Conditioned Architectural Volumetric Design Generation', in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 11936-11945. Available at: https://doi.org/10.1109/ICCV48922.2021.01174

Fricker, P., et al. (2007). 'Organised Complexity', in 25th ECAADe Conference Proceedings. Frankfurt Am Main (Germany), pp. 695-701.

Li, H., Zheng, Y., Wu, X. and Cai, Q. (2019) '3D Model Generation and Reconstruction Using Conditional Generative Adversarial Network', International Journal of Computational Intelligence Systems, 12(2), pp. 697-705.

Licklider, J. C. R. (1960) 'Man-Computer Symbiosis', IRE Transactions on Human Factors in Electronics, HFE-1(1), pp. 4-11.

Mo, K. et al. (2019) 'PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding', in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA: IEEE, pp. 909–918.

Nauata, N., Chang, K.-H., Cheng, C.-Y., Mori, G. and Furukawa, Y. (2020) 'House-GAN: Relational Generative Adversarial Networks for Graph-Constrained House Layout Generation', in Computer Vision – ECCV 2020, Vol. 12346, pp. 162-177. Springer International.

Oussidi, A. and Elhassouny, A. (2018) 'Deep generative models: Survey', in 2018 International Conference on Intelligent Systems and Computer Vision (ISCV). Fez: IEEE, pp. 1–8.

Palm, R., Paquet, U. and Winther, O. (2018) 'Recurrent Relational Networks', in Advances in Neural Information Processing Systems. Curran Associates, Inc . Available at:https://proceedings.neurips.cc/paper_files/paper/2018/file/b9f94c77652c9a76fc8a442748cd54bd-Paper.pdf

Wu, J., Zhang, C., Xue, T., Freeman, B., & Tenenbaum, J. (2016). 'Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling'. In Advances in Neural Information Processing Systems pp. 82-90. Available at: https://proceedings.neurips.cc/paper/2016/file/44f683a84163b3523afe57c2e008bc8c-Paper.pdf

Zhang, H. and Huang, Y. (2021) 'Machine Learning Aided 2D-3D Architectural Form Finding at High Resolution', in Proceedings of the 2020 DigitalFUTURES. Singapore: Springer, pp. 159–168.

Zhao, L. et al. (2020) 'T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction', IEEE Transactions on Intelligent Transportation Systems, 21(9), pp. 3848–3858.

Zhong, X., Fricker, P., Yu, F., Tan, C., & Pan, Y. (2022). 'A Discussion on an Urban Layout Workflow Utilizing Generative Adversarial Network (GAN) - With a focus on automatized labeling and dataset acquisition', in ECAADe 2022 Conference Proceedings. Ghent, 13-16 September 2022, pp. 583-592.