Kalpa Publications in Computing
Volume 22, 2025, Pages 756–769
Proceedings of The Sixth International Conference on Civil and Building Engineering Informatics

MULTI-STORY FLOOR PLAN GENERATION FROM BUILDING VOLUMETRIC DESIGN USING GRAPH NEURAL NETWORK

Atsuya Ogura[1], Tomohiro Fukuda[2], Nobuyoshi Yabuki[3], and Taro Narahara[4]

1) M.S. Student., Division of Sustainable Energy and Environmental Engineering, Graduate School of Engineering, Osaka University, Japan. Email: aogura0207yabuki@gmail.com
2) Ph.D., Assoc. Prof., Division of Sustainable Energy and Environmental Engineering, Graduate School of Engineering, Osaka University, Japan. Email: fukuda.tomohiro.see.eng@osaka-u.ac.jp
3) Ph.D., Prof., Division of Sustainable Energy and Environmental Engineering, Graduate School of Engineering, Osaka University, Japan. Email: yabuki@see.eng.osaka-u.ac.jp
4) Ph.D., Assoc. Prof., Hillier College of Architecture and Design (HCAD), New Jersey Institute of Technology (NJIT), Newark, NJ, USA. Email: taro.narahara@njit.edu

Abstract: The automatic design of architectural floor plans using deep learning has been widely studied to assist architectural design. Traditionally, floor plans generated by deep learning have been limited to single floors. Recently, research has been developing the use of graph neural networks (GNNs), which applies deep learning to graph data to generate building volumes that consider the in-building spatial use. Although these studies aim to generate new building volumes, practical architectural design often requires the generation of floor plans within predefined building outlines, constrained by various legal and regulatory requirements. This study proposes a method for generating multi-story floor plans in a given building volume by using a graph convolutional network (GCN), which adapts convolutional operations to graph data representing the given building volume. The implemented GCN model successfully predicted, with an accuracy of 74.66%, the spatial use class for each node within the graph representing the building. This enables the generation of detailed floorplans across multiple floors. This research contributes to the design support of multi-story floor plans in a given building volume. Moreover, when integrated with the latest 3D generative AI technologies, this approach promises to advance the automatic creation of 3D building models with comprehensive interior designs, starting from scratch in volumes initially devoid of any interior information.

Keywords: Floorplan generation, 3D building layout, Graph neural network, Deep learning, Generative design

1. INTRODUCTION

The design of building volumes and floor plans is conducted at the early stages of architectural design. Therefore, it is a crucial design process that significantly influences subsequent detailed architectural, equipment, and structural design (Zhong et al. 2023). Automated design of building volumes and floor plans enables the rapid presentation of diverse design alternatives to design decision makers. Additionally, from the diverse automatically generated designs, the optimal design can be selected based on various objective functions such as livability, environmental impact, building operability, and structural stability, which are significant motivations for automating the design of building volumes and floor plans (Weber et al. 2022).

The attempt to automate floor plan design using computers began in the early 1970s (Friedman 1971). Since then, there have been three main types of methods for automatic floorplan generation: 1) Bottom-up methods, 2) Top-down methods, and 3) Referential methods (Weber et al. 2022). Of these, Bottom-up methods and Top-down methods are antithetical methodologies, while Referential methods widely refer to methods that use machine learning and deep learning, and are sometimes applied to Bottom-up methods and Top-down methods.

Bottom-up methods generate floor plans by taking required space uses, such as offices and break rooms, as elements, connecting them, and aggregating these elements. The generated floor plans are evaluated based on the relationships between the rooms (Merrell et al. 2010) and environmental factors such as sunlight conditions (Yi and Yi 2014), thereby producing the most suitable floor plan for the intended objectives.

Top-down methods are used when designing within a predefined building envelope, particularly when architectural shapes are strongly constrained by building regulations or regional treaties, or when renovating existing structures. This method subdivides the input building envelope, such as building volume and floorplan outline, to generate a floor plan that best meets desired objectives (Baušys and Pankrašovaité 2005; Rodrigues et al. 2014).

Referential methods generate floor plans by referencing databases of past floor plans, similar to how architectural designers learn from past architectural works. Deep learning used for referential methods include models that learn image data of floor plans (Wu et al. 2019; Chaillou 2020) and models that learn graph data (Hu et al. 2020; Nauata et al. 2021). However, deep learning models that learn and generate image data have difficulty generating floor plans for multiple floors. This is because the dataset of floorplan image used for deep learning is a set of single floorplan images (Weber 2022). On the other hand, deep learning methods that learn graph data, which are called graph neural networks (GNNs), have prior studies, Building-GNN (Zhong et al. 2023) and Building-GAN (Chang et al. 2022), which generate multi-story floorplans.

Graph data refers to data consisting of a set of nodes and edges. On this graph data, GNNs perform tasks such as node classification, edge link prediction, and graph generation. Many types of data, including molecular structures, power networks, and social networks, can be described as graph data, and GNNs have been applied in various fields such as chemistry, pharmacology, and transportation (Zhou et al. 2020). Their application is also anticipated in the construction industry (Jia et al. 2023).

Both Building-GNN (Zhong et al. 2023) and Building-GAN (Chang et al. 2022), represent building volumes with spatial use information as graph data and use GNN to generate unknown building volumes. This is a study applying the Referential method to the Bottom-up method. However, no study has been proposed to apply GNNs to the Top-down method, which generates multiple floor plans from predefined building volumes.

Therefore, the objective of this study is to generate multi-story floorplans in a given building volume. This study proposed and implemented a method to predict the spatial use in a given building volume and generate floor plans for multiple floors by developing a GNN, specifically graph convolutional network (GCN), which extends convolutional operations to graph data (Kipf and Welling 2017) and predicts which spatial use a node in the graph belongs to.

This research contributes to design support for designing floor plans that span multiple floors within a given building volume. In addition, the proposed method of adding floor plans as a starting point for interior design to a building 3D model that does not have design information in the building volume is expected to contribute to the generation of building 3D models with detailed interior design when combined with the recently developed 3D generation AI.

2. RELATED WORK

2.1 Architectural Floorplan Generation Using GNNs

Previous studies applying deep learning to generate floorplans can be broadly classified into two types: methods that learn from image data and methods that use graph data as constraints.

In the methods of generating floorplans by training image data to deep learning, it is first required to unify the floorplan images of the dataset to be trained, such as 128 × 128 pixels or 256 × 256 pixels (Wu et al. 2019). This limits the scale of the floorplan to a relatively small scale. Furthermore, since the training images are single-floor plans, the generated floor plans are also restricted to being single-floor (Weber et al. 2022).

On the other hand, the method of generating floor plans by providing graph data to deep learning as a constraint is to first represent rooms as nodes and connect movable rooms with edges to represent the relationship between rooms in a graph (bubble diagram). The floorplan is then generated by evaluating the generated floorplan with the ground truth floorplan while maintaining the constraints between rooms in the bubble diagram, and then sequentially improving the floorplan to make it more similar to the ground truth floorplan. The floorplans can be generated bubble diagram, making it easier for users to generate the desired floorplans (Hu et al. 2020; Nauata et al. 2021).

In all these cases, there is no direct way to generate multiple-story floor plans. However, there are studies that use GNNs to learn the connection and arrangement of spatial uses in a 3D building model and generate building volumes that consider the connection and arrangement of spatial uses. In these studies, the generated building volume is cut horizontally, and this cut plane is used as a floor plan to create floor plans for multiple floors. The method proposed in this study also predicts the interior spatial use of a given building volume and generates a multi-story floor plan by cutting the building volume horizontally.

2.2 Building Volume to Multi-Story Floorplans

Common representation methods for building 3D models include voxel representation, point clouds, mesh representation, and neural fields (Wang et al. 2023). Of these, the representation method used by previous studies that apply GNNs to generate building 3D models is based on voxel representation.

Building-GNN (Zhong et al. 2023) generates the overall building volume by combining GNN and recurrent neural network, successively adding new building volumes based on the partial building volume initially provided by the designer. Both the initial building volume and the additional building volumes contain class information about the spatial use in the building. As a result, the final generated building model retains this internal usage information. By taking horizontal cross-sections of this building volume with usage information at each floor level, it is possible to obtain floor plans that span multiple floors.

Building-GAN (Chang et al. 2022) proposes a method to generate a new building volume by taking as input a bubble diagram of a floor plan spanning multiple floors and the designable area of the building. Building-GAN, like Building-GNN, generates a building volume with information on the building's internal spatial use, so that by obtaining a cut plane for each floor of this volume, a floor plan for multiple floors can be obtained. However, both Building-GNN and Building-GAN applied GNNs to bottom-up methods for floorplan generation, and there is no research on the application of GNNs to top-down methods for generating floorplans for multiple floors within a predefined building volume.

3. PROPOSED METHOD

This study proposes a method to automatically generate multi-story floor plans in a given building volume. The input is a building volume divided horizontally by reference lines and vertically by floor heights, and the output is multiple floor plans within the building volume. In the framework of this automatic floorplan generation, the main process consists of the three steps shown in Figure 1: Step 1. Conversion from Input Building Volume to Unclassified Voxel Graph; Step 2. Conversion from Unclassified Voxel Graph to Classified Voxel Graph using GNN; Step 3. Conversion from Classified Voxel Graph to Building Volume with Spatial Usage Class. By going through these three steps, the given building volume is converted into a building volume with spatial usage class. Multi-story floor plans are obtained by creating a sectional view of this building volume with classes of in-building spatial use.

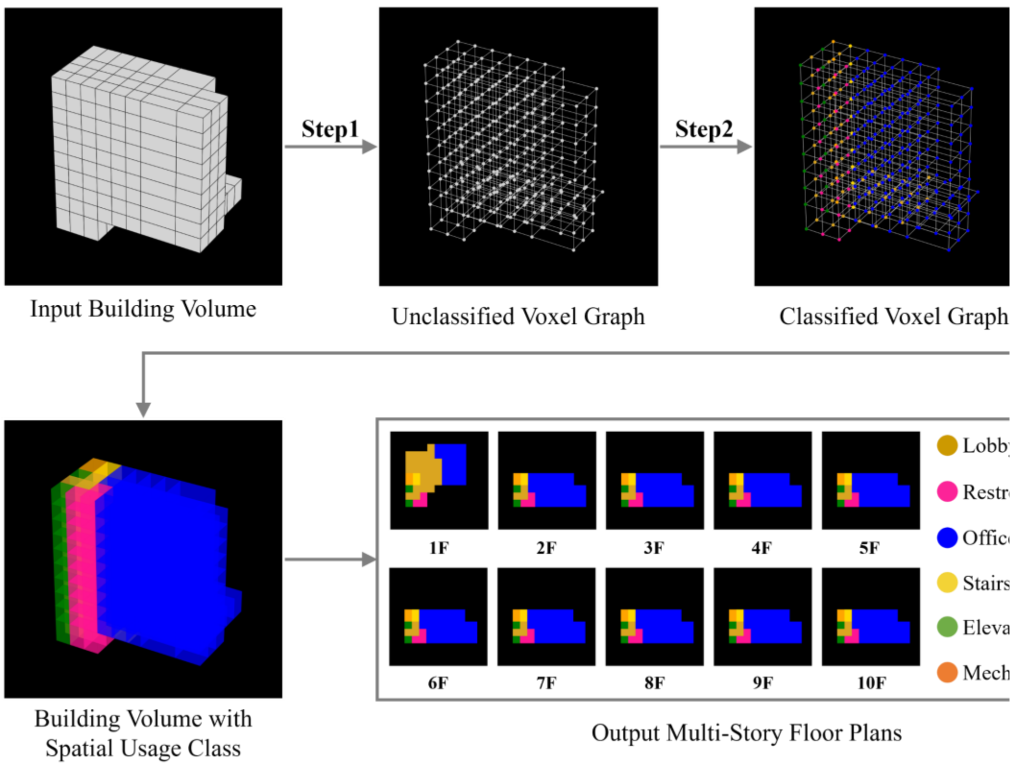
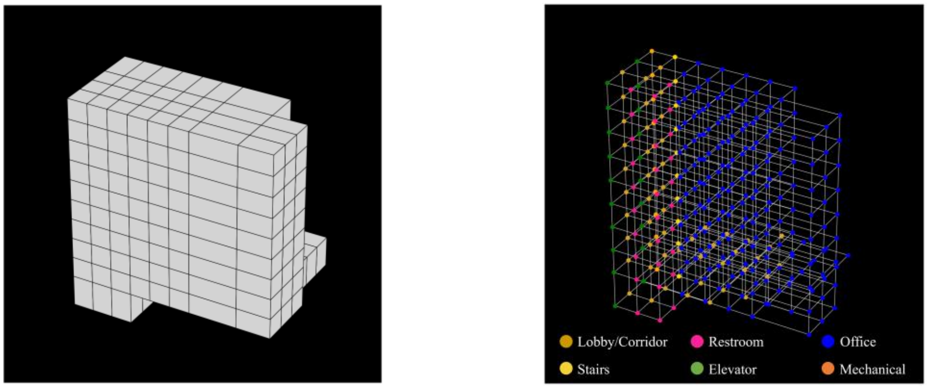


Figure 1. Overview of Proposed Method

3.1. Voxel Graph

Voxel graph is a method proposed by Building-GAN to represent building volumes as a 3D lattice graph (Chang et al. 2022). This 3D lattice graph is set up as follows. First, the voxel-represented building volume is divided by the reference lines used for design. The original building volume is then represented as a set of cuboids, as shown in Figure 2 (a). A 3D lattice graph is obtained by taking the center points of these cuboids and connecting the lattice points with edges when the cuboids share a face with a neighboring cuboid. Voxel graph is a graph in which these grid points are nodes, and the node's attribute information includes the node's coordinates, the node's class information (the class that represents the spatial use in the building), and the height, width, and height of the building voxel that a node represents. This study used the dataset of voxel graph provided by Chang et al. (2022), which is a dataset of 120,000 office buildings generated according to patterns and rules provided by professional architects. Figure 2 (b) shows a voxel graph where each node has either six classes information (Lobby/Corridor, Restroom, Office, Stairs, Elevator, and Mechanical).



(a) (b)

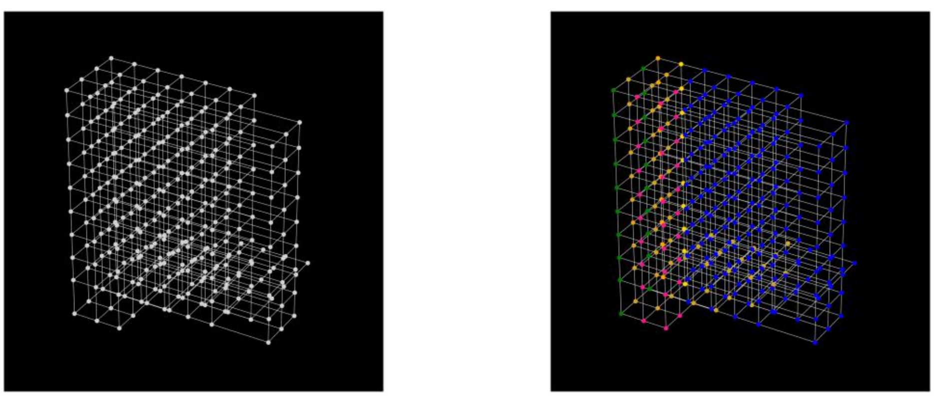
Figure 2. (a) Building Volume. (b) Voxel Graph.

3.2. Conversion from Input Building Volume to Unclassified Voxel Graph

As shown in Figure 1, Step 1 converts an architectural volume into an unclassified voxel graph.

(1) Unclassified Voxel Graph and Classified Voxel Graph

In this study, a voxel graph whose nodes do not have in-building spatial use class information is called an unclassified voxel graph (Figure 3 (a)), and a voxel graph whose nodes have in-building spatial use class information is called a classified voxel graph.



(a) (b)

Figure 3. (a) Unclassified Voxel Graph. (b) Classified Voxel Graph.

(2) Building Volume with Any Representation Methods to Unclassified Voxel Graph

In the method proposed in this study, a given building volume refers to the volume and shape of the space occupied by a building. The representation method of the building volume is not specified. The method of representation can be voxel representation, mesh representation of the volume surface, volume rendering, or any other arbitrary method. When a building volume created using any representation method is divided by the design baseline, the building volume is divided into a set of cuboids, regardless of the method of representation. By obtaining nodes at the center points of these cuboids, an unclassified voxel graph can be obtained.

3.3. Conversion from Unclassified Voxel Graph to Classified Voxel Graph using GNN

In Step 2, the unclassified voxel graph is converted to a classified voxel graph using GNN. Step 2 is a process to predict which class of spatial use a node without class information in the unclassified voxel graph belongs to and is a node class classification problem in GNN.

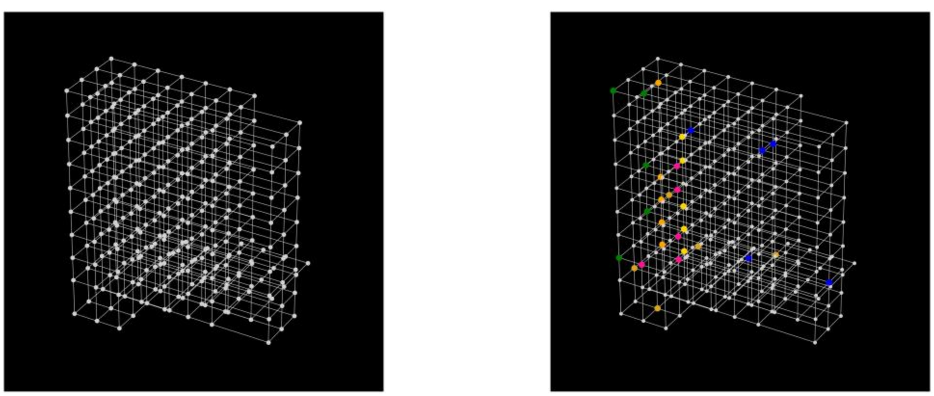
(1) Dataset

For GNN to learn the voxel graph as a node classification problem, this study created a voxel graph dataset with the configuration shown in Table 1. The number of voxel graphs used for training data is 10,000. The number of voxel graphs used for validation data and test data is both 1,000. A single voxel graph has three types of data: 1) adjacency matrix, 2) node features, and 3) node's ground truth class label.

Table 1. Structure of Voxel Graph Dataset

The number of training data	10,000
The number of validation data	1,000
The number of test data	1,000
Data in one voxel graph	Adjacency matrix
	Node feature (Coordinate or Signal)
	Ground truth class label for each node

For the node features, the following two cases were tested: 1) the X, Y, and Z coordinates were used as node features (Coordinate Feature); 2) some nodes were given the ground truth class label (Signal Feature), which was used as node features. Each of these cases can be interpreted in the voxel graph, as shown in Figure 4.



(a) (b)

Figure 4. (a) Coordinate Feature.
(b) Signal Feature. 5 nodes for each class are given ground truth labels in this case (5 signals).

When GNN is trained with X, Y, and Z coordinates as node features, the GNN will predict the class of in-building spatial use (e.g., office, facility room, etc.) from the given building volume alone. When some of the nodes are given the ground truth class label, and GNN is trained, the GNN will classify the remaining nodes that were not given the ground truth label. This is equivalent to the designer specifying the spatial use of a part of the building volume and using it as a constraint, and the GNN automatically designs the rest of the in-building spatial use.

(2) Graph Neural Network Architecture

The basic idea of GNNs is to update the feature vector of each node with the feature vectors of its neighboring nodes, thereby enriching the node's features to features that reflect the structure of the graph. Equation (1) is the update mechanism for a typical GNNs.

$$\mathbf{h}_i = \sum_{j \in \mathcal{N}_i} x_j W^T \tag{1}$$

Where x_j is the feature vector for node j. $\mathbf{h}_i$ is the updated vector for node i. W is the weight vector. $\mathcal{N}_i$ is the set of node i and its neighboring nodes.

This study used GCN (Kipf and Welling 2017) to classify nodes. General GNN models have the problem that nodes with large degree (the degree of a node is the number of edges that incident to it) have extremely large influence compared to nodes with small degree due to the difference in the degree of the nodes. As shown in Equation (2), GCN normalizes the imbalance that general GNNs suffer from by dividing the updated features of a node by degrees when updating the features of the node.

$$\mathbf{h}_i = \frac{1}{\deg(i)} \sum_{j \in \mathcal{N}_i} x_j W^T \tag{2}$$

Where $\deg(i)$ is the degree of node i.

The GNN that learns the voxel graph dataset and performs node class classification has the configuration shown in Figure 5: Node features of the voxel graph are embedded, passed through the 4-layer GCN, and connected to all coupling layers to perform classification for each node.

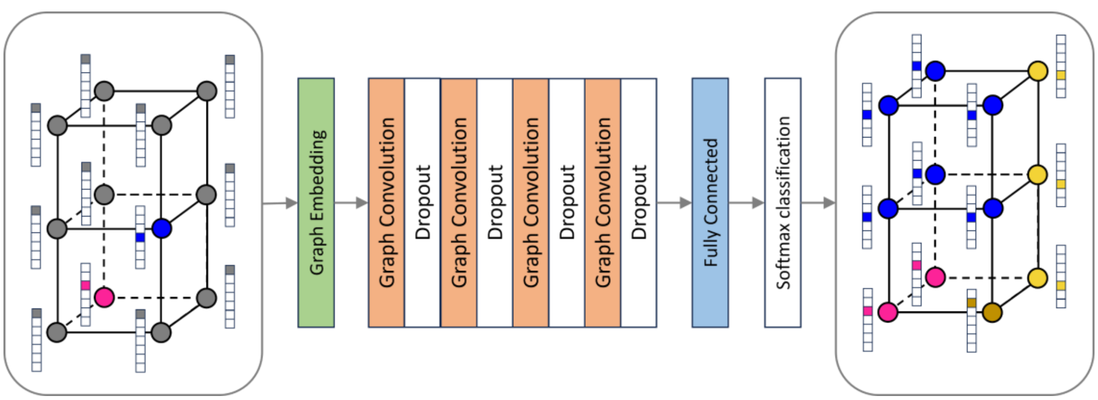


Figure 5. Structure of Graph Neural Network for Voxel Graph Dataset

3.4. Conversion from Classified Voxel Graph to Building Volume with Spatial Usage Class

In Step 3 in Figure 1, the classified voxel graph obtained by Step 2 is converted into a building volume with class information on in-building spatial use. In Step 3, the classified voxel graph can be converted into a building volume by voxelizing each node of the classified voxel graph according to the design baseline, centered at the node's coordinates, in the reverse of Step 1. Finally, the building volume with in-building spatial use is cut horizontally according to the reference lines to output floor plans for each floor.

4. RESULTS

4.1. Implementation Details

The GNN used in this study was implemented with Pytorch and Deep Graph Library (Wang et al. 2019), which provides a Python package necessary for GNN implementation. The Optimizer is AdamW, the initial value of the learning rate is 5×10^{-5}, and if the loss function decreases for 25 epochs, the learning rate is halved, and when the learning rate falls below 1×10^{-6}, the GNN training is completed, and the node classification accuracy at this time is defined as classification accuracy of the GNN implemented in this study.

4.2. Definition of Node Classification Accuracy

The classification of voxel graph node by GNN was performed using 10,000 voxel graphs as training data, with coordinate feature and signal feature. The overall node classification accuracy is the average accuracy for each class. The accuracy for one class is obtained by dividing the number of nodes whose ground truth label was correctly estimated by GNN by the number of nodes belonging to that class. The node classification accuracy for each class is expressed in Equation (3), and the overall node classification accuracy is expressed in Equation (4).

$$Accuracy\ for\ Class\ k\ (A_k) = \frac{N_{Pred\ (k)}}{N_{GT\ (k)}} \tag{3}$$

$N_{GT\ (k)}$ is the number of nodes whose ground truth label is class k. $N_{Pred\ (k)}$ is the number of nodes that is predicted to belong to class k by GNN.

$$Accuracy = \frac{\sum_{k=1}^{nb_{classes}} A_k}{nb_{classes}} \tag{4}$$

4.3. Node Classification Accuracy

The relationship between node classification accuracy and the epochs on the training data is illustrated in Figure 6. The "5 signals" scenario represents the case where five nodes per class are randomly selected in advance, and their ground-truth classes are provided as node features. Similarly, the "10 signals" scenario corresponds to the case where ten nodes per class are randomly selected, with their ground-truth classes used as node features. The node classification accuracy for both the training and test datasets is summarized in Table 2. For the test dataset, when the node features were XYZ coordinates, the accuracy was 22.8%. In the "5 signals" scenario, the accuracy increased to 56.67%, and in the "10 signals" scenario, it further improved to 74.66%.

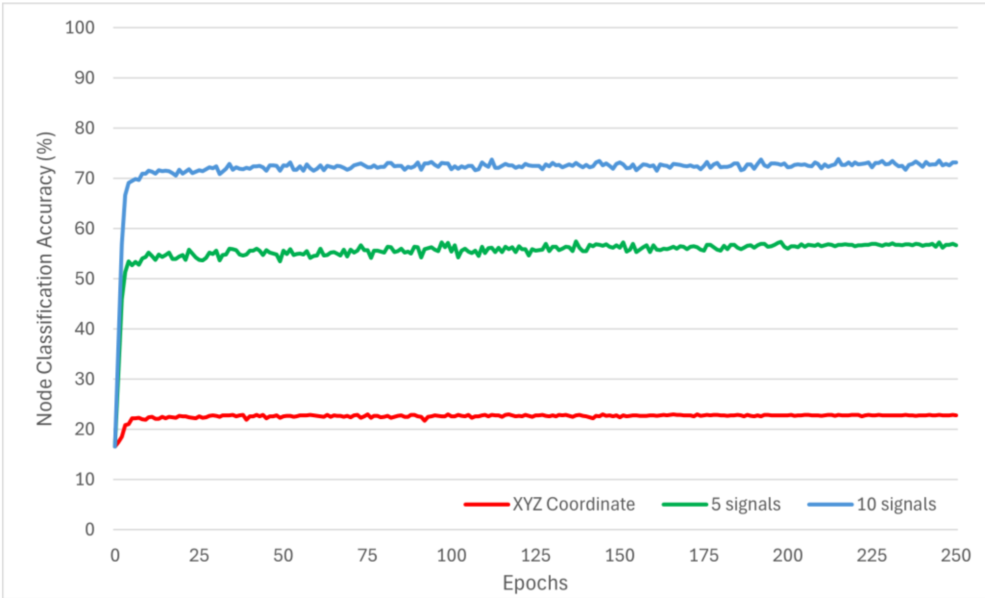


Figure 6. The Relationship between Node Classification Accuracy and the Epochs on the Training Data

Table 2. Node Classification Accuracy on Test Data and Training Data

	XYZ Coordinate	5 Signals	10 Signals
Test Data	22.81%	56.67%	74.66%
Training Data	22.35%	56.37%	76.01%

The multi-story floor plans are directly obtained from the building volume that is voxelized from the classified voxel graph by Step 3 in Figure 1. Therefore, this GNN node classification accuracy, which indicates the accuracy of Step 2 of generating the classified voxel graph, is an indicator of how well the floorplan generated predicts the ground truth floorplan of the input building volume.

4.4. Output Floorplans

Figure 7 shows the results of floorplan generation by integrating the GNN trained by 5 signals and 10 signals, respectively, into the proposed method. The input building volume in Figure 7 is an example of a building volume randomly extracted from the test data of Voxel Graph Dataset.

	Ground Truth	Generated Floorplans with 5 Signals	Generated Floorplans with 10 Signals

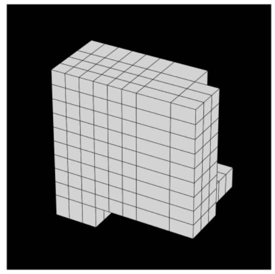
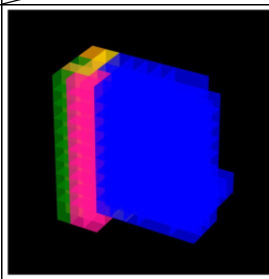
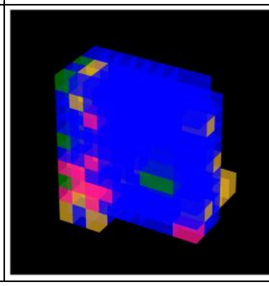
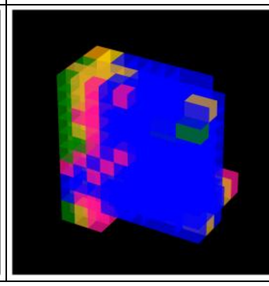
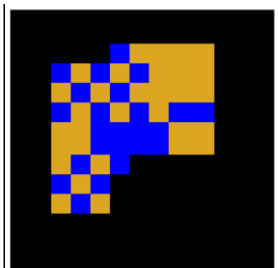
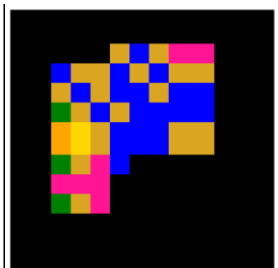
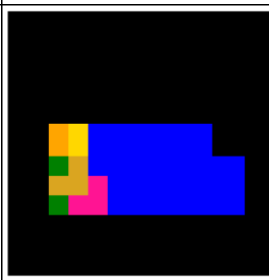
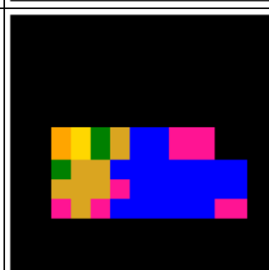
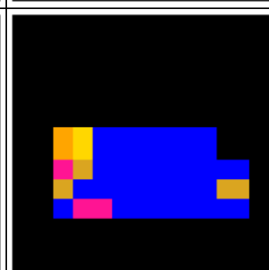
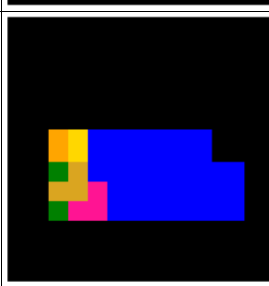
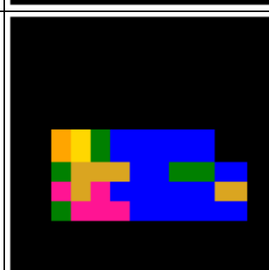
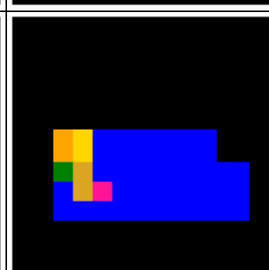
Input Building Volume			Lobby/Corridor Restroom Office Stairs Elevator Mechanical
Node Classification Accuracy		56.67 %	74.66 %
Generated Building Volume			

Figure 7. Result of Generated Floorplans (continued)

1F			
2F			
3F			

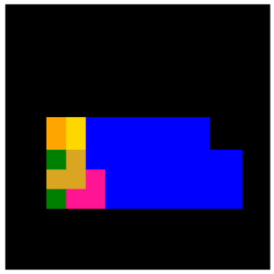
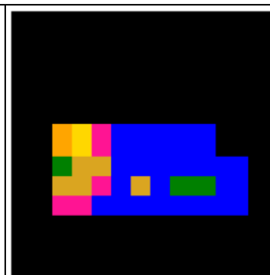
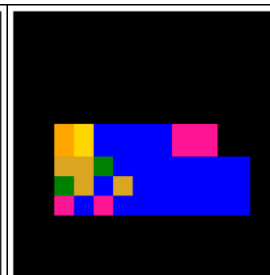
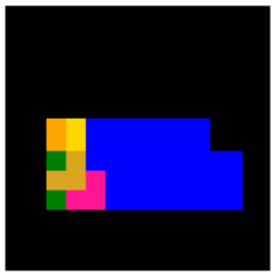
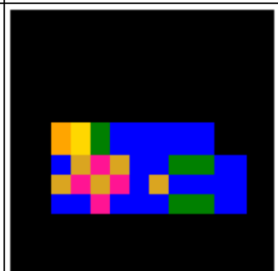
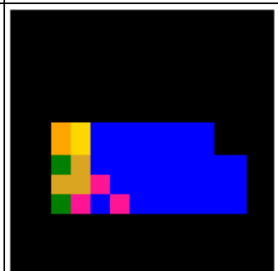
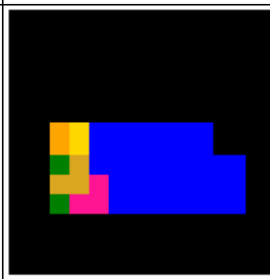
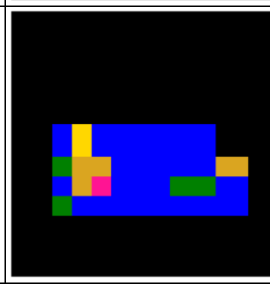
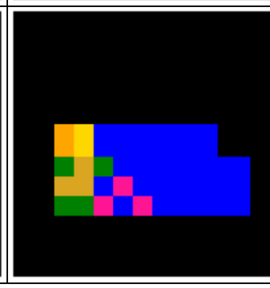
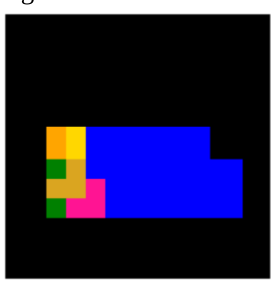
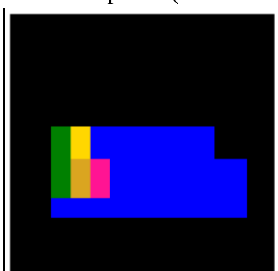
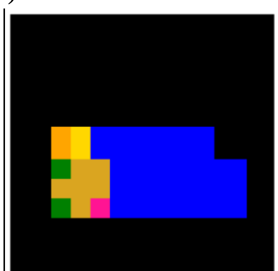
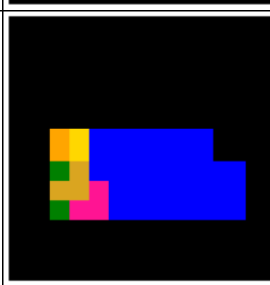
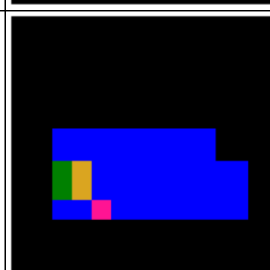
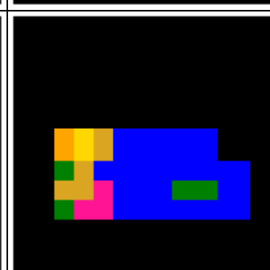
4F			
5F			
6F			

Figure 7. Result of Generated Floorplans (continued)

7F			
8F			

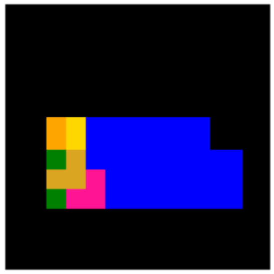
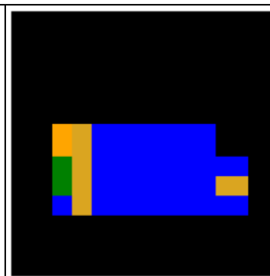
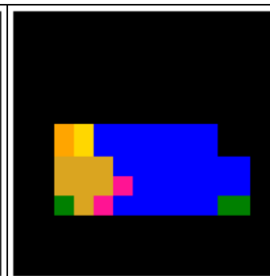
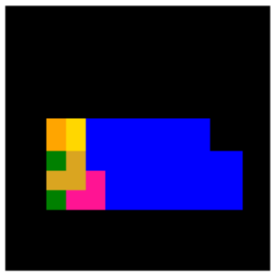
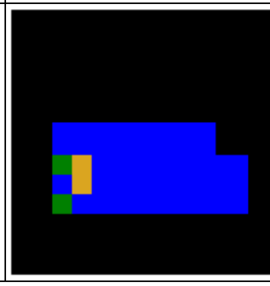
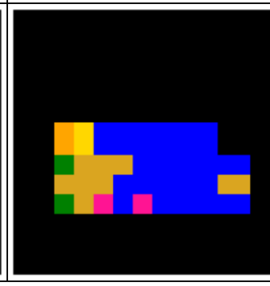
9F			
10F			

Figure 7. Result of Generated Floorplans

5. DISCUSSION

Comparing the generated floor plans with the Ground Truth floor plans, the generated first floor plans do not replicate the Ground Truth floor plans as well as the floor plans on the second and higher floors. This can be attributed to the fact that the Ground Truth floor plans differ significantly between the first floor and the second floor and above. This suggests that GNN was able to learn the features of similar floor plans on the second and higher floors of Ground Truth but was unable to learn the rapidly changing floor plan on the first floor. In addition, when looking at the floor plans generated when five signal features were given, there is a horizontal series of green areas indicating elevators. This may be improved by adding coordinate features to the five signal features and training the GNN. In addition, the voxel graph dataset used in this study was designed for office buildings, so the generation of floorplans by this study is limited to office buildings. Different datasets should be prepared for the design of hotels, high-rise residential buildings, and commercial facilities.

6. CONCLUSION

This study proposed a novel method to generate multi-story floor plans in a given building volume by using GCN on voxel graph, which represents building volumes with spatial use information as graph data. The GNN implemented in this study predicts the remaining volume's spatial uses from a small number of spatial uses given as initial design conditions. By specifying 5 nodes per class for each spatial use in the voxel graph, the remaining nodes are classified with 56.67% accuracy. If 10 nodes per class for each spatial use are specified within the voxel graph, the remaining nodes are classified with 74.66% accuracy.

The contributions of this study are as follows.

- A novel method was proposed to generate multi-story floor plans within a given building volume.
- It has been demonstrated that a GNN can be implemented to predict the spatial use of the remaining building volume by specifying the spatial use for a part of the building volume.

The proposed method generates a floor plan, serving as the starting point for subsequent interior design within a 3D building model that initially contains no internal details. Future work will aim to integrate this research with 3D generative AI that creates building volumes, potentially enabling the generation of 3D building models with detailed interior designs.

ACKNOWLEDGMENTS

This work was partially supported by JSPS KAKENHI Grant Number 23K11724.

REFERENCES

Baušys, R., & Pankrašovaité, I. 2005. "Optimization of architectural layout by the improved genetic algorithm." *Journal of Civil Engineering and Management*, 11(1).

Chang, K.-H., Cheng, C.-Y., Luo, J., Murata, S., Nourbakhsh, M., & Tsuji, Y. 2022. "Building-GAN: Graph-Conditioned Architectural Volumetric Design Generation." *IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 11936-11945.

Friedman, Y. 1971. "Flatwriter: choice by computer." *Progressive Architecture* 03 89–101.

Hu, R., Huang, Z., Tang, Y., van Kaick, O., Zhang, H., & Huang, H. 2020. "Graph2Plan: Learning Floorplan Generation from Layout Graphs." *ACM Transactions on Graphics*, 39(4).

Jia, Y., Wang, J., Shou, W., Hosseini, M. R., & Bai, Y. 2023. "Graph neural networks for construction applications." *Automation in Construction* (Vol. 154).

Kipf, T. N., & Welling, M. 2017. "Semi-supervised classification with graph convolutional networks." 5th *International Conference on Learning Representations, ICLR* 2017 - Conference Track Proceedings.

Merrell, P., Schkufza, E., & Koltun, V. 2010. "Computer-Generated Residential Building Layouts." *ACM Transactions on Graphics*, 29(6).

Nauata, N., Hosseini, S., Chang, K. H., Chu, H., Cheng, C. Y., & Furukawa, Y. 2021. "House-GaN++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Rodrigues, E., Gaspar, A. R., & Gomes, Á. 2014. "Automated approach for design generation and thermal assessment of alternative floor plans." *Energy and Buildings*, 81.

Wang, A., Dong, J., Lee, L., Shen, J., & Hui, P. 2023. "Towards AI-Architecture Liberty: A Comprehensive Survey on Designing and Collaborating Virtual Architecture by Deep Learning in the Metaverse." *ArXiv*.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., & Zhang, Z. (2019). Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. ArXiv. /abs/1909.01315

Weber, R. E., Mueller, C., & Reinhart, C. 2022. "Automated floorplan generation in architectural design: A review of methods and applications." *Automation in Construction* (Vol. 140).

Wu, W., Fu, X. M., Tang, R., Wang, Y., Qi, Y. H., & Liu, L. 2019. "Data-driven interior plan generation for residential buildings." *ACM Transactions on Graphics* (TOG), 38(6), 1-12.

Yi, H., & Yi, Y. K. 2014. "Performance based architectural design optimization: Automated 3D space layout using simulated annealing." *2014 ASHRAE/IBPSA-USA Building Simulation Conference.*

Zhong, X., Koh, I., & Fricker, P. 2023. "Building-GNN: Exploring a co-design framework for generating controllable 3D building prototypes by graph and recurrent neural networks." *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).

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. 2020. “Graph neural networks: A review of methods and applications.” *AI Open* (Vol. 1).

Zhou, P., & Chang, Y. 2021. “Automated classification of building structures for urban built environment identification using machine learning.” *Journal of Building Engineering*, 43, 103008.