BRIDGING BIM AND AI:

A Graph-BIM Encoding Approach For Detailed 3D Layout Generation Using Variational Graph Autoencoder

JIADONG LIANG[1], XIMING ZHONG[2] and IMMANUEL KOH[3]
[1]*Architectural Association School of Architecture.*
[2]*Aalto university school of art design and architecture.*
[3]*Singapore University of Technology and Design.*
[1]*liangjiadongarch@gmail.com, 0009-0005-9244-8674*
[2]*ximing.zhong@aalto.fi, 0000-0002-3060-6981*
[3]*immanuel_koh@sutd.edu.sg, 0000-0002-1181-1082*

Abstract. Building Information Modelling (BIM) data provides an abundant source with hierarchical and detailed information on architectural elements. Nevertheless, transforming BIM data into an understandable format for AI to learn and generate controllable and detailed three-dimensional (3D) models remains a significant research challenge. This paper explores an encoding approach for converting BIM data into graph-structured data for AI to learn 3D models, which we define as Graph-BIM encoding. We employ the graph reconstruction capabilities of a Variational Graph Autoencoder (VGAE) for the unsupervised learning of BIM data to identify a suitable encoding method. VGAE's graph generation capabilities also reason for spatial layouts. Results demonstrate that VGAE can reconstruct BIM 3D models with high accuracy, and can reason the entire spatial layout from partial layout information detailed with architectural components. The primary contribution of this research is to provide a novel encoding approach for bridging AI and BIM encoding. The Graph-BIM encoding method enables low-cost, self-supervised learning of diverse BIM data, capable of learning and understanding the complex relationships between architectural elements. Graph-BIM provides foundational encoding for training general-purpose AI models for 3D generation.

Keywords. BIM, Graph-Structured, Encoding Method, VGAE, Graph Reconstruction and Generation

1. Introduction

The current challenges in AI learning of 3D models stem from a lack of high-quality 3D datasets and the complexities of encoding 3D models to capture spatial layout features. Building Information Modelling (BIM) provides an abundant source, including 3D scanning and design and construction data. BIM encompasses detailed and hierarchical information on architectural elements, such as geometric data, spatial relationships, and material properties. Nevertheless, transforming BIM data into an understandable format for AI to generate controllable and detailed 3D models remains a significant research challenge(Bassir et al., 2023).

Encoding refers to transforming raw data into a format suitable for processing and learning by deep learning models. In this paper, the encoding method we discuss involves converting three-dimensional architectural models into data for AI learning. Typical methods of 3D model representation such as point clouds, voxels, and neural fields, can be encoded to facilitate AI learning(Tang et al., 2022; Zhong et al., 2023; Mildenhall et al., 2022). These methods primarily focus on the spatial attributes of individual elements but struggle to learn the interrelationships between elements within spatial layouts and face challenges in complex architectural spatial arrangements (Zhong et al., 2023).

We propose an encoding approach that transforms BIM data into graph-structured data for AI to learn 3D models, referencing encoding based on graph representation to align with the data structure of the BIM model, which we define as Graph-BIM. Graph-structured data, comprised of nodes and edges, serve to represent relationships between entities (Xu et al., 2018). Concurrently, BIM data includes detailed information about each architectural element, such as walls, doors, and windows, and spatial relationships between these elements. These two data structures exhibit inherent compatibility. In applying BIM data to graph-structured data, architectural elements are encoded as nodes. The physical connections, spatial relationships, or functional dependencies among these components are conceptualized as edge. This graph-structured data enables AI to capture the details of each architectural element and to articulate the interrelationships between these elements and the spatial layout. Recent studies have demonstrated the potential of graph-structured encoding methods in learning architectural 3D spatial layouts (Nauata et al., 2020; Zhong et al., 2023). However, they are beset with two principal issues. Firstly, encoding approaches from the aforementioned studies are constrained in the diversity of learnable data types, with irregular architectural spatial layouts posing significant challenges to learning. Secondly, these encoding methods predominantly rely on room function bubble diagrams or space voxels, limiting their direct control and generation capabilities for architectural components such as columns and walls. In contrast, Graph-BIM facilitates AI's comprehension and learning of multiple types of BIM data and reconstructs and generates precise, controllable detailed BIM models. To validate the feasibility of the Graph-BIM, we select the Variational Graph Auto-Encoders (VGAE) within the Graph Neural Networks (GNNs) framework for the experiment.

GNNs lie in their capability to process graph-structured data, effectively learning the intricate relationships among nodes(Xu et al., 2018). For instance, tools such as shear wall analysis tools have demonstrated the capability of GNNs to learn graph-structured data and reason spatial layouts (Zhao et al., 2023). However, GNNs in the aforementioned studies require extensive labelling of data during training. To facilitate more efficient learning of BIM data, we select the VGAE model. Comprising an encoder and decoder, VGAE facilitates self-supervised learning of BIM data, capturing the nodes' complex features and their topological interrelations within the graph, thereby generating a new graph (Kipf & Welling, 2016). Recent studies have demonstrated the successful application of VGAE in graphically representing molecular structures, enabling prediction edge generation and edge attributes and generating novel molecular structures (Bresson & Laurent, 2019). This suggests that VGAE also has the potential to learn BIM models represented as graph-structured data, understanding the relative spatial relationships between different architectural components. Therefore, we aim to explore suitable Graph-BIM encoding methods and predict building layouts by assessing the accuracy of spatial reconstructions by VGAE. Figure 1 illustrates the overall application process of Graph-BIM and the employed VGAE model, where users obtain detailed Revit models by inputting building outlines and inner control points.

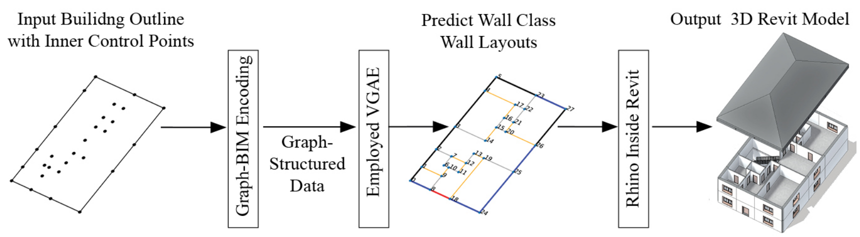

Figure 1. User Interface for 3D Layout Generation Using Graph-BIM and Employed VGAE

The principal contribution of the Graph-BIM encoding approach is to bridge AI and BIM. It empowers AI with the ability to self-supervise efficiently in the learning of BIM data features, facilitating cost-effective learning across multiple types of BIM-3D datasets. Graph-BIM encoding enables AI to grasp the relationship between architectural elements and to generate detailed 3D models.

2. Related Work

Nauata et al. (2020) develop House-gan, a system for generating housing layouts that comply with graph constraints. In House-gan, rooms are represented as nodes and their adjacencies are depicted as edges in the graph structure. Its abstraction oversimplifies the representation of detailed architectural elements such as columns, walls, and windows. Building-GNN encodes model voxels as nodes and their spatial interrelations as edges, allowing for precise spatial semantics control at the voxel level(Zhong et al., 2023). However, it faces challenges in learning irregular architectural layouts and generating architectural components within the voxel space. A GNNs-based method for shear wall layout prediction encodes structural elements like walls and windows as edges, with nodes representing component intersections(Zhao et al., 2023). This approach accurately predicts shear wall layouts, demonstrating the potential for architectural element control and prediction. Nevertheless, it primarily focuses on shear wall attributes, not encompassing the prediction of the overall spatial layout. The limitations in these studies in detailing architectural features, learning data types, and reason overall spatial layouts led to the development of the Graph-BIM encoding approach. Our method encodes architectural components such as columns and walls as graph nodes and edges, assigning them coordinates and specific positional attributes relative to the entire layout. Furthermore, to understand relationships among architectural elements, we propose incorporating room function connections as special attributes within our encoding framework.

Simonovsky and Komodakis (2018) demonstrate the use of GraphVae to output a predefined probabilistic fully connected graph in a single step using the decoder. By setting probability thresholds, it controls the likelihood of the existence of nodes and edges, thereby generating molecular graphs. This inspires us to test multiple threshold values to retain more complete reconstructed results of spatial layouts. The research of Permutation-Invariant Variational Autoencoder focuses on graph-level representation learning, introducing a novel model that addresses the graph reordering conundrum, and illustrating the management of node sequence uncertainty within graph structures(Winter et al., 2021). This provides specific technical guidance for handling the node sequence in various architectural layouts using VGAE. Shi et al. (2020) introduce a masking label prediction strategy in training the Message Passing Model, which involves randomly masking a certain proportion of input label information before making predictions. This inspires us to hide parts of the spatial layout information to reason the overall space, testing whether the Graph-BIM encoding allows the VGAE model to accurately learn the spatial features. Given these theoretical explorations and practical applications, we resolve to utilize the VGAE model's graph reconstruction capabilities to test suitable Graph-BIM encoding methods, while also exploring the potential of the VGAE model for reasoning architectural spatial layouts using Graph-BIM encoding BIM data.

Our research aims are to employ the VGAE model for reconstructing architectural spatial layouts and test appropriate Graph-BIM encoding methods, and to utilise VGAE 's graph generation capabilities to infer comprehensive architectural spatial layouts from localized information and generate new architectural spatial layouts.

3. Methodology

The Graph-BIM encoding experiment workflow is shown in Figure 2.

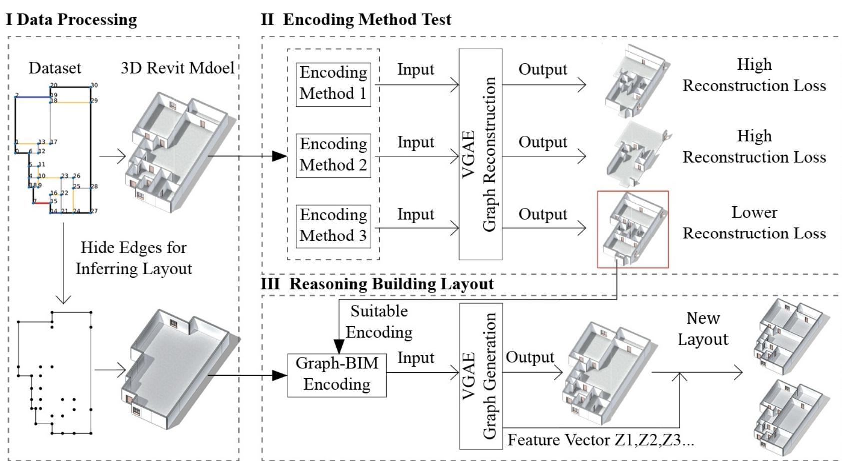

Figure 2. Workflow of Graph-BIM Encoding and VGAE Experiment

3.1. ENCODING METHOD

We encoded the endpoints of walls as nodes and non-overlapping columns were also encoded as nodes, with node features including spatial location and environmental information(Nauata et al., 2020; Zhao et al., 2023; Zhong et al., 2023). Walls were encoded as edges, with edge attributes representing wall types. We combined these three encoding methods and validated them using the VGAE model(Figure 3).

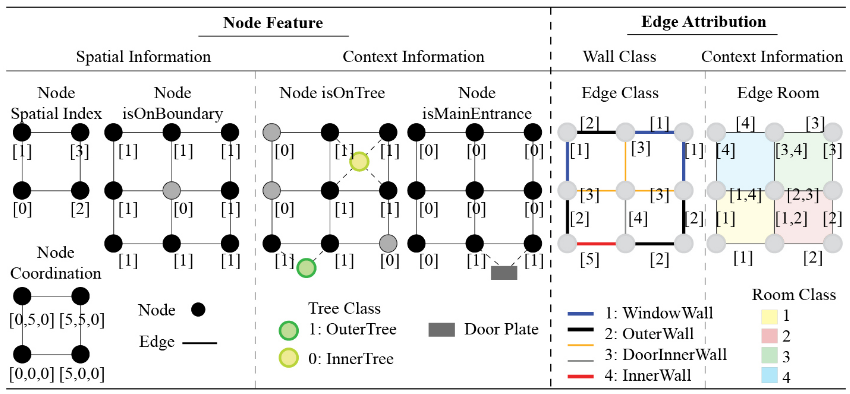

Figure 3. Encoding Methods

3.2. DATA PROCESSING

In this experiment, we utilized two datasets: the House-gan-MainEntrance (House-gan -ME) dataset and the Tree-Grid dataset(Figure 4). The House-gan -ME dataset was chosen for its variety in node numbers and irregular shapes, satisfying the requirements for datasets with diverse node quantities and non-uniform forms(Nauata et al., 2020). In addition to columns and wall endpoints, the Tree-Grid dataset included nodes representing environmental information. This approach facilitated the verification of the Graph-BIM encoding's accuracy in data format transformation for BIM data, unrestricted by the number of data nodes, node types, and layout shapes. We obtained BIM data through Revit to simulate the real workflow of AI learning BIM models.

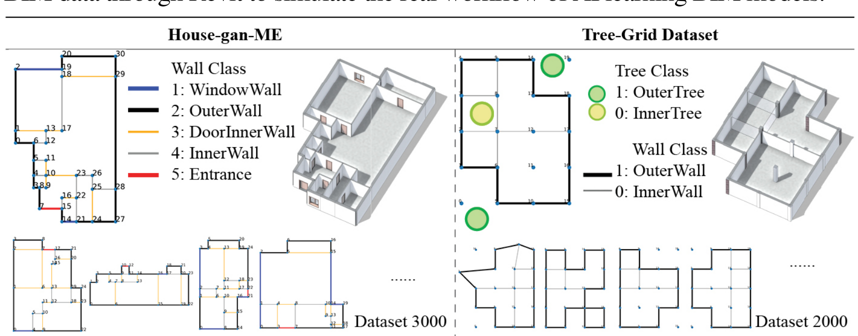

Figure 4. Dataset for VGAE Training

3.3. GRAPH RECONSTRUCTION AND GENERATION USING VGAE

Initially, the VGAE model underwent self-supervised graph reconstruction training. The output, consisting of the reconstructed adjacency matrix (positions of edges) and edge attributes, was compared with the original data's adjacency matrix and edge attributes to calculate the loss(Figure 5(a)). We conducted experiments with three different encoder models on three encoding methods to test suitable Graph-BIM encoding approaches. Then, we used this Graph-BIM encoding method to train the VGAE model's spatial reasoning capabilities. We concealed some of the walls(Edges) in datasets, retaining only the outermost walls of the building. The VGAE model outputted building layouts, which were compared with complete datasets for loss calculation(Figure 5(b)). Finally, we verified the capability to generate new layouts based on the Graph-BIM encoding. By using the VGAE to reconstruct different spatial layout feature vectors and linearly combining them in various proportions to create new feature vectors, the VGAE decoded these vectors into adjacency matrices and edge attributes. This generated result contained layout features proportionate to those in the original architectural space, as represented by the feature vectors.

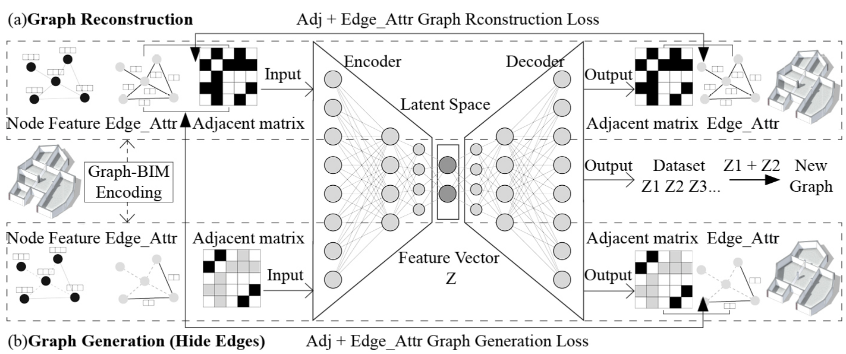

Figure 5. VGAE Graph Reconstruction and Generation

4. Result

4.1. RESULTS OF AD-GAN GENERATION AND DESIGN DECISIONS

The experimental results indicated that the VGAE model using Encoding Method 3 exhibits the best reconstruction capability. In the TransformerGCN model, the reconstructed wall positions and categories were almost identical to the original data. Observing the loss on the test set, Encoding Method 3 demonstrated the lowest reconstruction loss of spatial layouts across all three encoder models, with an average loss on the test set shown in Figure 6 being 0.098. This suggested that Encoding Method 3 enabled the VGAE to accurately learn the relationships between architectural components and spatial layouts. The reconstruction results of Encoding Method 2 were the least effective, either generating many irrelevant diagonal walls or missing most wall information. In the GINE model experiment, the reconstruction loss on the test set was as high as 0.756. This implied that relying solely on environmental information encoding was insufficient for effectively learning the finer details in BIM data. The reconstruction loss of Encoding Method 1 was close to that of Encoding Method 3, capable of reconstructing wall layouts and categories more completely. In the TransformerGCN experiment, the reconstruction loss decreased to 0.184, with only a minimal number of walls missing. In the GINE model experiment, the reconstructed spatial layout was missing some wall information, with a reconstruction loss of 0.574. This indicated that spatial information played an important role in encoding methods, but environmental information was still necessary to learn local features. Specific encoding contents on the Tree-Grid and House-gan-ME datasets were presented in Figure 7.

	TestDataset Mean Loss for Adj + Edge Attr Reconstruction		
Encoder Model	Encoding Method 1 Spatial Information +Edge Attribution	Encoding Method 2 Context Information +Edge Attribution	Encoding Method 3 Spatial+Context Information +Edge Attribution
GCN Kipf & Welling, 2016	0.227	0.613	0.117
GINE Xu et al., 2018	0.574	0.756	0.431
TransformerGCN Shi et al., 2020	0.184	0.412	0.098

Encoding Method 1	Encoding Method 2	Encoding Method 3
Origin Reconstruction 3DModel	Origin Reconstruction 3DModel	Origin Reconstruction 3DModel

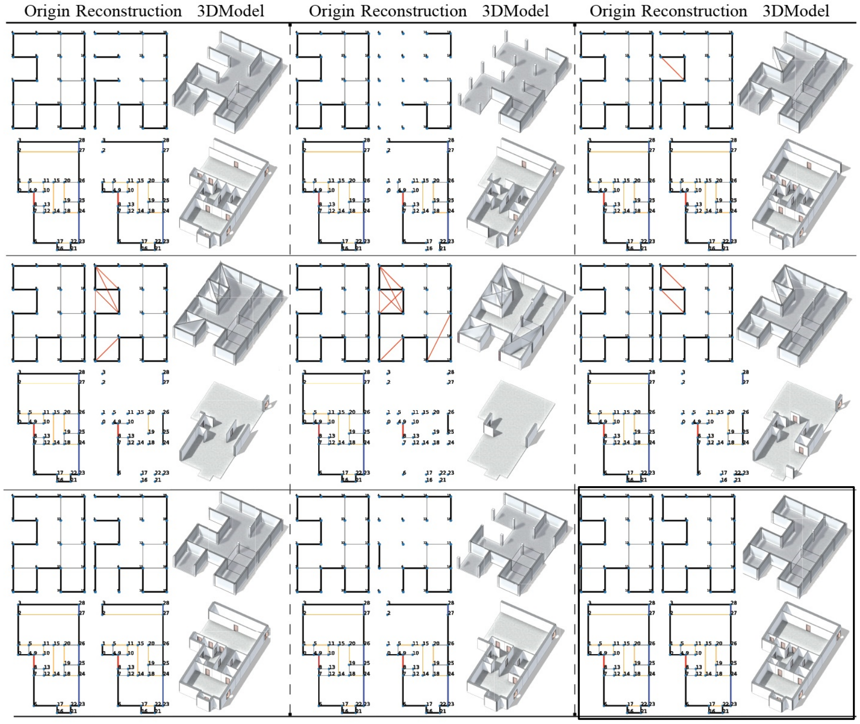

Figure 6. Graph Reconstruction Results for Testing Encoding Method

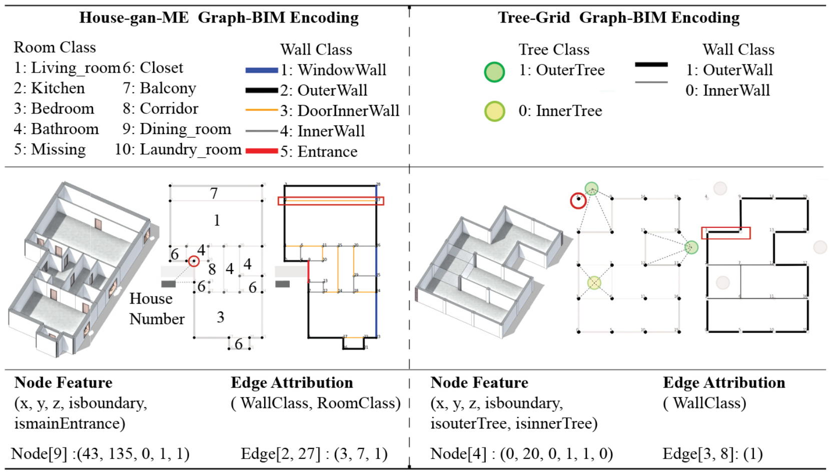

Figure 7. Graph-BIM Encoding Details

4.2. REASONING SPATIAL LAYOUT RESULTS

Figure 8 illustrated that the VGAE model employing the Graph-BIM encoding method was capable of reasoning complete spatial layout results on the test dataset based on partial layout information, and the spatial layouts satisfactorily met environmental constraints. The positions and categories of walls closely resembled the original spatial layout. The walls corresponding to door plates were precisely those of the main entrance category, with the lowest loss compared to the original at 0.067 (Figure 8(a)). Figure 8(d) demonstrated that both the external balconies and internal courtyards in the layout corresponded well with trees, with a reconstruction loss of 0.0743. Although some walls were lost in Figures 8(e) and (f), the impact on the overall spatial layout was not significant. The results substantiated the generalization capability of the Graph-BIM and VGAE could accurately generate spatial layouts that complied with environmental constraints for both regular and irregular spatial arrangements.

4.3. NEW LAYOUTS GENERATION RESULTS

Figure 9 showed the new layouts generated by concatenating feature vectors of different layouts. At a specific vector ratio of 0.3:0.7, the architectural layout tended more toward the feature2 layout, reducing a wall and forming a longitudinal spatial layout(Figure 9(a)). At a 0.5:0.5 ratio, the new spatial layout embodied the spatial characteristics of both original schemes, featuring both horizontal and longitudinal wall layouts, as depicted in Figure 9(b). When the ratio is 0.7:0.3(Figure 9(c)), the layout features leaned more towards feature 1, with an additional wall generated at the bottom of the space due to the influence of feature 2, yet the overall layout remained predominantly horizontal. The experimental results demonstrated that VGAE was capable of generating new building layouts from the combination of different feature vectors. This contributed to data sample augmentation and promoted diversity in the design.

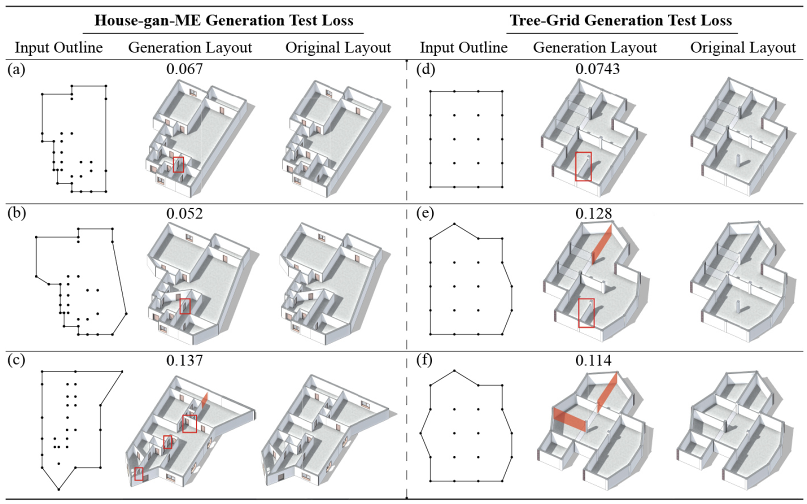

Figure 8. Graph Generation Results of Reasoning Building Layout

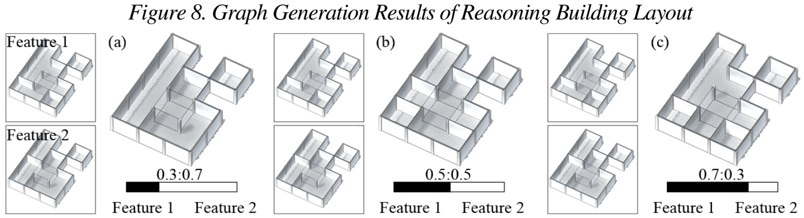

Figure 9. New Building Layouts Generation by Combining Feature Vectors

4.4. LIMITATION

The datasets primarily consist of spatial layouts formed by walls and columns. To enhance AI's performance in learning layouts, it is advisable to incorporate a wider range of architectural components, such as windows, doors, and furniture, into the Graph-BIM encoding method. This expansion aims to facilitate the learning and generation of more comprehensive 3D models. During the experiments, we employed three types of graph convolutional networks, which effectively aggregate node information. However, the integration of edge attributes into node features was suboptimal. A neural network model capable of precisely aggregating edge attributes and node features would be instrumental in deepening the machine's understanding of spatial layouts. Currently, in the graph generation phase, we only reconstructed the adjacency matrix and edge attributes of the dataset, predicting walls based on the original graph's nodes. We will train AI to develop the capability to generate nodes, enabling it to infer and generate overall spatial layouts with minimal information.

5. Conclusion

This paper explores a Graph-BIM encoding approach for converting BIM data into graph-structured data for AI training. The main contribution is to provide a bridge between AI and BIM encoding, enabling self-supervised learning of BIM data, allowing AI to learn multiple types of BIM datasets at a low cost and surpassing previous encoding methods in understanding the complex relationships between architectural elements and generating detailed 3D models. Our work is beneficial for training general-purpose AI models for 3D generation. Utilizing the Graph-BIM, architects can input outlines and interior control points to generate detailed models in Revit. Future developments may enable the training of a universal AI agent capable of addressing design requirements through the Graph-BIM. This offers a paradigm of end-to-end human-machine collaborative design for both architects and machines.

Acknowledgements

We would like to acknowledge author1 and author2 contributed equally to this work.

References

Bassir, D., Lodge, H., Chang, H., Majak, J., & Chen, G. (2023). Application of artificial intelligence and machine learning for BIM: Review. International Journal for Simulation and Multidisciplinary Design Optimization, 14, 5. https://doi.org/10.1051/smdo/2023005

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Kipf, T. N., & Welling, M. (2016). Variational Graph Auto-Encoders (arXiv:1611.07308).

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2022). NeRF: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1), 99–106. https://doi.org/10.1145/3503250

Nauata, N., Chang, K. H., Cheng, C. Y., Mori, G., & Furukawa, Y. (2020). House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16 (pp. 162-177). Springer International Publishing.

Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., & Sun, Y. (2020). Masked label prediction: Unified message passing model for semi-supervised classification. arXiv preprint arXiv:2009.03509. https://doi.org/10.48550/ARXIV.2009.03509

Simonovsky, M., & Komodakis, N. (2018). Graphvae: Towards generation of small graphs using variational autoencoders. In Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27 (pp. 412-422).

Tang, S., Li, X., Zheng, X., Wu, B., Wang, W., & Zhang, Y. (2022). BIM generation from 3D point clouds by combining 3D deep learning and improved morphological approach. Automation in Construction, 141, 104422. https://doi.org/10.1016/j.autcon.2022.104422

Winter, R., Noé, F., & Clevert, D. A. (2021). Permutation-invariant variational autoencoder for graph-level representation learning. Advances in Neural Information Processing Systems, 34, 9559-9573. http://arxiv.org/abs/2104.09856

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How Powerful are Graph Neural Networks? https://doi.org/10.48550/ARXIV.1810.00826

Zhao, P., Fei, Y., Huang, Y., Feng, Y., Liao, W., & Lu, X. (2023). Design-condition-informed shear wall layout design based on graph neural networks. Advanced Engineering Informatics, 58, 102190. https://doi.org/10.1016/j.aei.2023.102190

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. In International Conference on Education and Research in Computer Aided Architectural Design in Europe (pp. 431-440). eCAADe.