A Cross-Modal AI Approach to Spatial Layout Generation: Fusing Graph, Text, and Boundary

Ximing ZHONG [1] , Jiadong LIANG [2], Xianchuan MENG [3]

Abstract: This paper introduces an innovative cross-modal spatial layout generation approach that leverages AI-agents to seamlessly integrate graphs, textual descriptions, and geometric boundary constraints for the creation of 3D room layouts. The proposed method utilizes a unique agent-based framework that incorporates large language models (LLM) and graph neural networks (GNN) to process and fuse multimodal inputs, allowing for a more comprehensive and flexible design process. By combining textual descriptions with boundary constraints into room feature embedding, the method enhances the semantic consistency and practicality of the generated layouts. Compared to the previous single-modal generative models, the experimental results demonstrate the method's effectiveness in accurately reconstructing room layouts and adapting to user-defined changes, showcasing its potential to revolutionize the field of architectural design by enabling the efficient integration of AI-agents and cross-modal data processing. Overall, this paper presents a significant step forward in the development of intelligent, adaptable, and user-centric tools for 3D spatial layout generation.

Keywords: Cross-modal；LLM；GNN；AI-agent；Spatial layout；3D Generation

1 Introduction

The three-dimensional (3D) spatial layout in architectural design is a complex task, due to the multiple objectives that influence the overall performance of the design outcome. Therefore, the spatial layout process in conventional architectural design is a dynamic, iterative, and time-consuming endeavour, constantly adjusted until the result with the best overall performance[1]. Location anchors (such as bubble diagrams), natural language descriptions of spatial attributes, and expected boundary of the design domain are extremely useful information in conventional design as essential reasoning information during the design process[3]. Fusing these different types of information is one of the challenges faced by AI based technology.

2 Related Works

In the field of artificial intelligence, cross-modal typically refers to the ability of machine learning models to process and understand different types of data. This includes tasks like converting visual information (images) into text descriptions or transforming text into 3D models. Advances in cross-modal AI technologies like GANs, ChatGPT, and Codex[2] have opened possibilities for architectural design by incorporating other modal inputs that align better with conventional design practices. Cross-modal technologies can integrate textual descriptions, 2D images, and 3D data to make decisions for complex tasks and generate more accurate 3D models[11]. This cross-modal approach can provide more convenient, efficient, and time-saving innovative potential for room layout design, making it more in line with the habits of conventional design.

Text-to-3D generation is a crucial cross-modal technology in spatial generation methods. CG3D adopts a two-stage approach to generate 3D from text, involving structured relationship generation and physics-based

[1] X. Zhong (✉), Aalto University, Otaniementie 14, 02150 Espoo, Finland, e-mail: ximing.zhong@aalto.fi

[2] J. Liang (✉), Architectural Association School of Architecture, 36 Bedford Square, London, United Kingdom, e-mail: liangjiadongarch@gmail.com

[3] X. Meng (✉), Nanjing Univeristy, 22 Hankou Road, Nanjing, China, e-mail: mxc@nju.edu.cn

optimization, aiming to address issues like text-controlled multi-object layouts, physical realism of scene compositions, and semantic and physical consistency[12]. MeshGPT generates triangle meshes directly as sequences of triangles, encoding triangle embeddings into latent quantized embeddings through a graph convolutional encoder and ResNet decoder, utilizing GPT to produce high-quality triangle sequences[10]. 3DMIT enhances LLMs' understanding of 3D scenes through end-to-end fine-tuning, injecting 3D information directly into LLMs and utilizing a 3D perceiver architecture with multiple encoders to extract and align global and detailed features in 3D scenes[6]. GALA3D utilizes large language models to guide initial layouts, then incorporates conditional diffusion priors to obtain more realistic and accurate real-world spatial layouts, ensuring spatial distribution and geometric consistency through adaptive geometry control and optimizing multi-object interactions in 3D scenes[16]. Text2Room proposes a method to generate complete, texture-rich 3D indoor scene meshes from textual inputs, first creating indoor layouts and furniture, then filling in gaps with 3D geometry, ensuring seamless and natural textures through depth alignment and mesh fusion[4]. However, these methods primarily record 3D models as meshes, with rendering as their primary task, lacking spatial topological relationships necessary for dynamic spatial layout design processes and adapting to architects' flexible design adjustments.

The topological relationships of subspaces are crucial for spatial plan generation methods. Data-driven approaches have initially demonstrated the effectiveness and controllability of architectural space generation. House-GAN++ is an intelligent framework for generating and optimizing floor plan layouts, representing room layouts as graphs, with nodes representing rooms and edges representing relationships between them, generating layouts based on input conditions like room count, types, and areas[9]. Zheng and Petzold decouple topological and geometric predictions of room layouts, using subgraph neural networks to predict room topologies and combining neural-guided plan sketching to generate layouts with reasonable topologies[13]. However, these methods focus primarily on data-driven generation from graphs to graphs, lacking other modal inputs, limiting the flexibility of on-demand adjustments during room generation. Spatial topological graph relationships overly abstract spatial requirements, whereas other modalities, such as boundary constraints or texts can more precisely store expected detailed information necessary for generation.

Boundary constraints are a type of geometric modality input that represents the design domain desired by the designer. Graph neural networks (GNN) can effectively integrate boundary constraints with topological constraints. Graph2Plan integrates user-in-the-loop design constraints based on given boundaries and bubble graphs. The system incorporates a retrieve-and-adjust paradigm, utilizes a deep neural network for floorplan generation, and includes a post-processing step for aligning room boundaries and vectorizing the floorplan[5]. Building-GNN emphasizes the controllability of the generative process that meets boundary conditions. Designers can traditionally define volumetric boundaries for the GNN model to infer volumetric details in space. Additionally, this method provides a flexible interactive design interface, realized through the GH-Python framework on the Rhino modelling platform, enabling architects to engage in a feedback loop with AI[14]. Utilizing building outlines and specified points as inputs, Graph-BIM is introduced, which encodes BIM data into graph structures tailored for AI learning, and helps AI understand and learn BIM data, enabling the generation of controllable 3D building models that meet spatial boundary constraints[7]. The results indicate that GNN models excel at generating vector models. Thus, we also adopted the previously advantageous GNN generation framework to explore architectural generation.

However, a common challenge with these methods is the need for human understanding to convert requirements into input features, as AI is unable to transform requirements into the features of input models across different modalities. For instance, in Building-GNN, architects cannot input text to modify the desired outcomes. In contrast, AI-agent technologies like the framework proposed by Building-Agent can intelligently convert textual requirements into 3D structured data, thereby reasoning to generate 3D layouts[15]. This enables us to use agents for data-driven feature engineering to inspire further development.

We propose a cross-modal spatial layout generation method that takes graph diagrams, such as bubble diagrams, and textual descriptions as inputs. This method utilizes an LLM-based agent framework and GNN to generate semantic 3D spatial layouts that meet boundary requirements and enables textual input to adjust room layout details during the generation process (Fig. 1). The way AI-agent integrates different modal information resembles the flexible process of conventional room layout design.

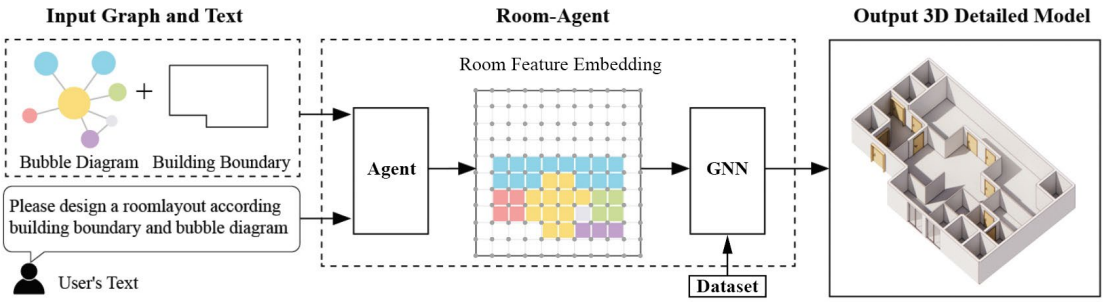

Fig. 1 Application of Room-Agent in architectural workflow

The contribution of this study lies in a cross-modal feature fusion method: utilizing various modalities such as spatial topological maps of mouse clicks, constraint boundaries, and text as input objects. Through the intelligent integration enabled by the AI-agent framework, this approach allows data-driven models to generate detailed 3D models that meet the dynamic evaluation needs of architects while conforming to the intricate layout features of datasets. Compared to earlier single-modal AI models like House-GAN, the inclusion of textual input enhances the model's ability for dynamic intelligent generation. Furthermore, the control through boundary constraints and graph anchoring ensures its potential for practical application in real projects.

3 Methodology

As shown in Fig. 2, the framework of the Room-Agent method is divided into three parts: 1) input representation, 2) Agent-driven feature engineering, and 3) GNN generation model. We initially employ graphs, text, and 3D boundaries as inputs, followed by utilizing multiple AI agents to facilitate the feature engineering process between the input and the GNN generation model. Specifically, the encoding unifies the requirements of different modalities into the features of graph nodes and edges. Finally, these features are fed into a trained GNN model, outputting the node and edge features of the graph, which are transformed into a new three-dimensional room layout. Additionally, text, anchor points, and boundaries can be used concurrently in the generation process for flexible adjustments to the layout, constituting an end-to-end modification process.

In our approach, the bubble diagram and building boundary are transformed into text using AI agents that automatically reads and processes data formats to meet the requirements of structured data (Fig. 3). The graph and boundary vector information are converted into structured text output for accurate processing and efficient use of diverse data by another specialized agent that handles graph and boundary data. The "bubble diagram to text" agent function demonstrates how to convert a bubble diagram that represents rooms with coordinates and areas into a text format, while the "building boundary to text" agent function illustrates the method of translating building boundaries into text. Based on these structured data, room features first maps the coordinates of different room types within the building boundary, then colours the potential diffusion areas of functional rooms using Signed Distance Function (SDF), overlaying the diffusion area with a grid for evaluation, and calculates the influence probability of graph and text inputs on the properties of the final layout nodes, achieving feature embedding as input for AI generative models. By integrating features from different modalities into unified embedded features, AI agents provide more comprehensive and accurate inputs for generative models.

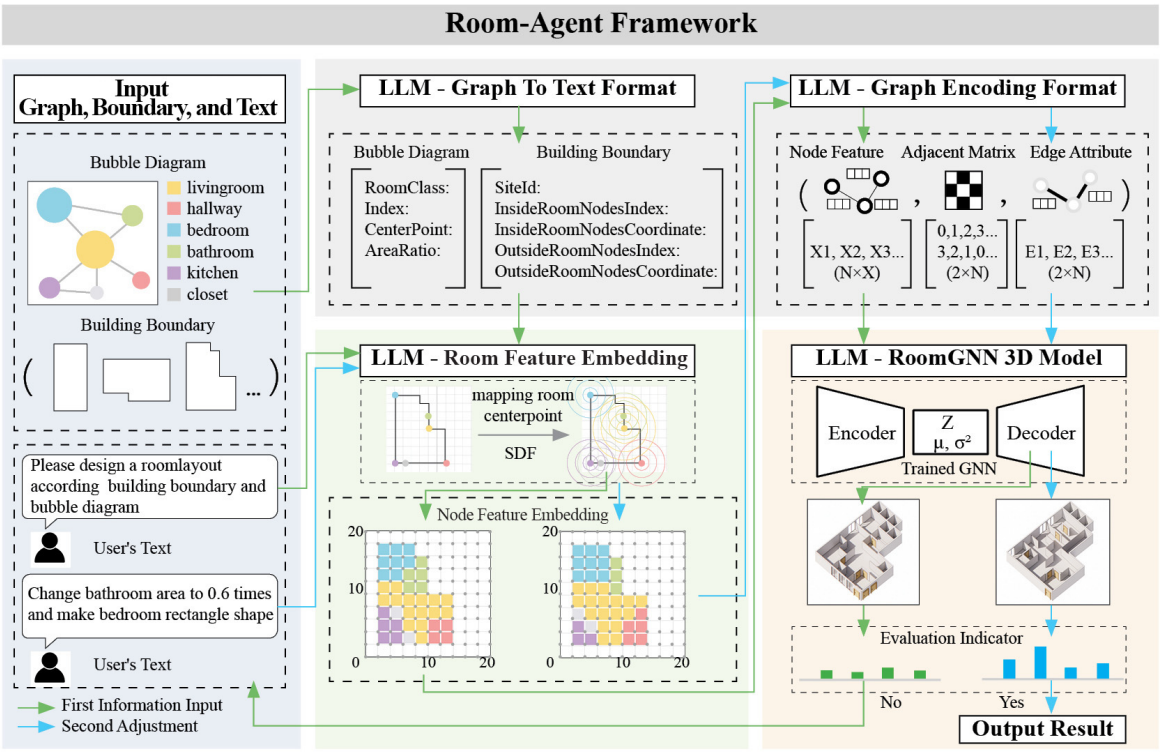

Fig. 2 Room-Agent framework of converting graph, boundary, and text to 3D detailed model

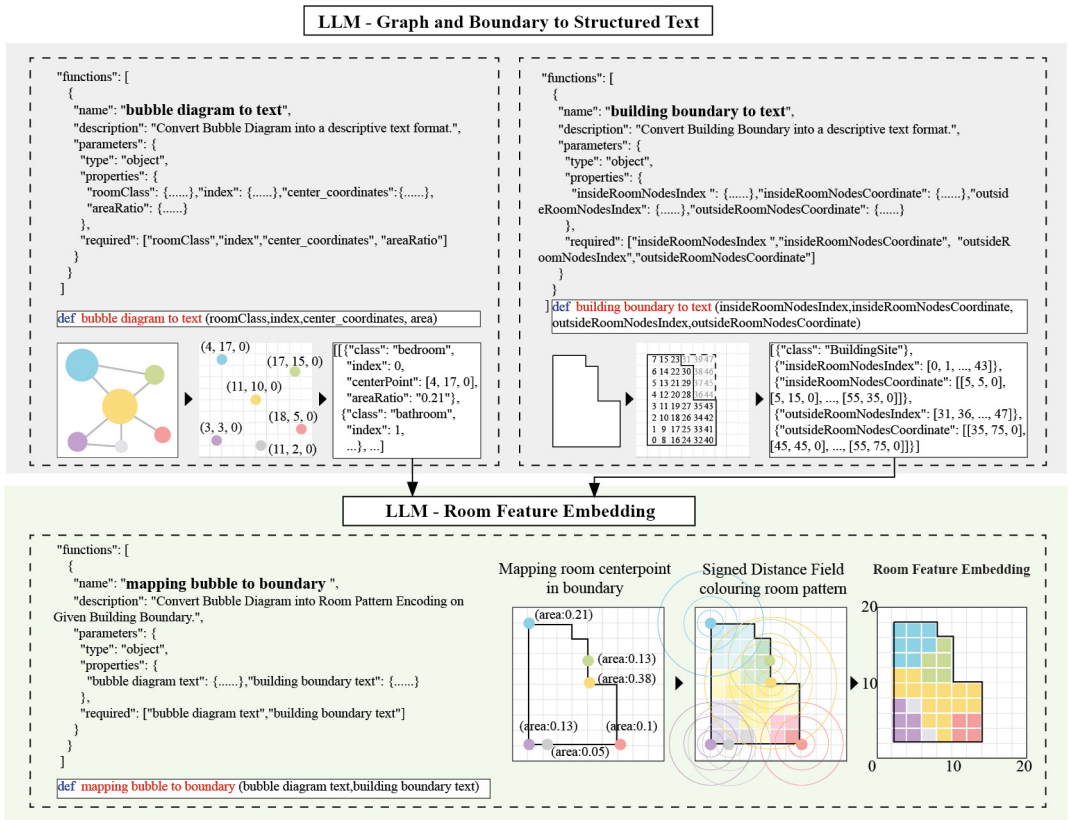

Fig. 3 LLM functions converting graph, boundary, and text to room feature embedding

GNN is employed as a generative model in this approach. Fig. 4 demonstrates the details of the training architecture and generative process of RoomGNN. Fig. 5 shows how graphs represent the details of 3D models. The model training process inputs node data encoded based on room feature embedding, while the model input consists of edge attributes. Loss is calculated by comparing the predicted results with the edge categories in the dataset layout, enabling the GNN model to learn the features between node inputs and spatial layouts, thus predicting the details of the spatial layout, as shown in the input and output sections of Fig. 5. Once the data training is completed, the feature embedding derived from user input is fed into the trained model to predict models. From the test dataset, we can observe the model's generalization and generative capability. Since we have set corresponding 3D components to different edge types, we can directly obtain the 3D model of the room layout.

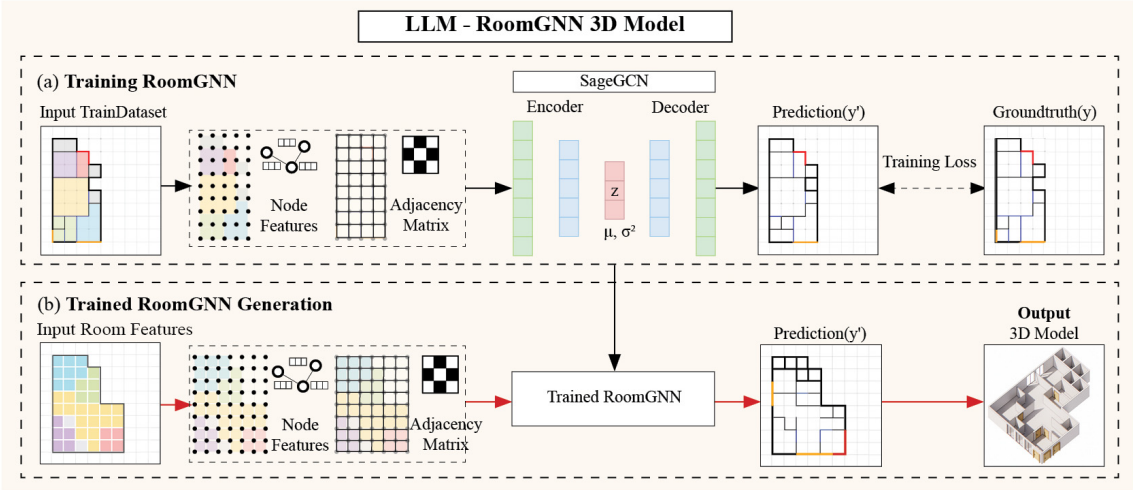

Fig. 4 Room-GNN dataset and graph encoding method

Fig. 5 illustrates the details of how a 3D model is encoded into a graph data structure that AI can understand. In simple terms, nodes represent spatial grids, while edges indicate the categories of walls. The node features include properties such as coordinates, boundary types, and room categories. Edges are represented by an adjacency matrix defined by endpoints, and edge types are encoded using a one-hot encoding scheme to represent various wall types. This encoding method has been shown in previous studies to be superior to other encoding techniques for improving AI predictions of layout details[8]. During training, edge types are used for cross-entropy loss calculations, where the loss function reflects the discrepancies in spatial layout details. Therefore, a reduction in loss directly indicates effective learning.

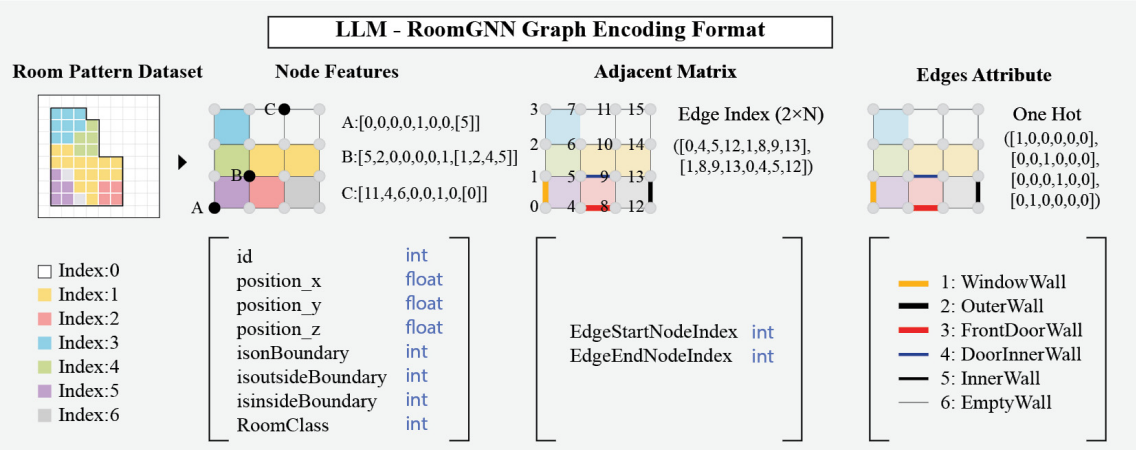

Fig. 5 The encoding method for the spatial layout generation

As shown in Fig. 6, the AI model of RoomGNN is trained using the Housegan-Grid dataset, which is a modified version of the Housegan dataset. This dataset samples walls within a composite grid layout, offering precise delineation of room functions and various wall construction types.

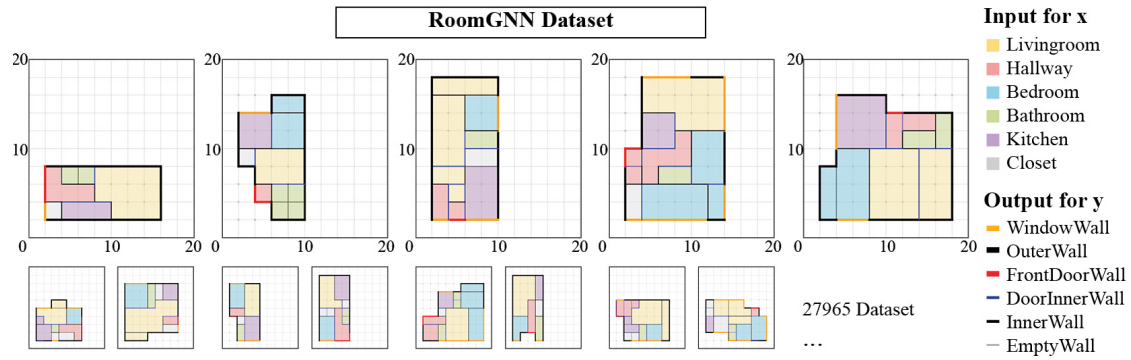

Fig. 6 Dataset used in Room-Agent

4 Experiment and Result

The results of RoomGNN on the test set, compared to the ground truth, demonstrate excellent learning capability, enabling the inference of complete room layouts from simplified room divisions. The average loss for the model's graph reconstruction results on the test set is 0.41 (Fig. 7(b)), as the loss itself represents the deviation of each layout-generating element from the dataset, showing its ability to effectively capture features related to spatial layouts within the dataset. The loss values of the generated results in Fig. 7(a)(1)(3)(4)(8)(9)(11) are significantly lower than the average, with the minimum loss at 0.207, indicating an almost complete reconstruction of the original spatial layout. This showcases RoomGNN's outstanding reasoning ability in translating room functions into detailed spatial layouts. However, when faced with complex architectural boundaries, the loss of RoomGNN's reconstructed spatial layouts tends to be higher. Notably, in Fig. 7(a)(2)(5)(6)(10)(12), the maximum loss reaches 0.565. Some reconstruction results exhibit additional walls both inside and outside the architectural boundaries, affecting the usability of the rooms, and some windowed exterior walls were not accurately predicted. Overall, RoomGNN's reconstruction performance on the test set highlights the effectiveness of this room-encoding approach, successfully learning layout logics such as wall types and spatial divisions.

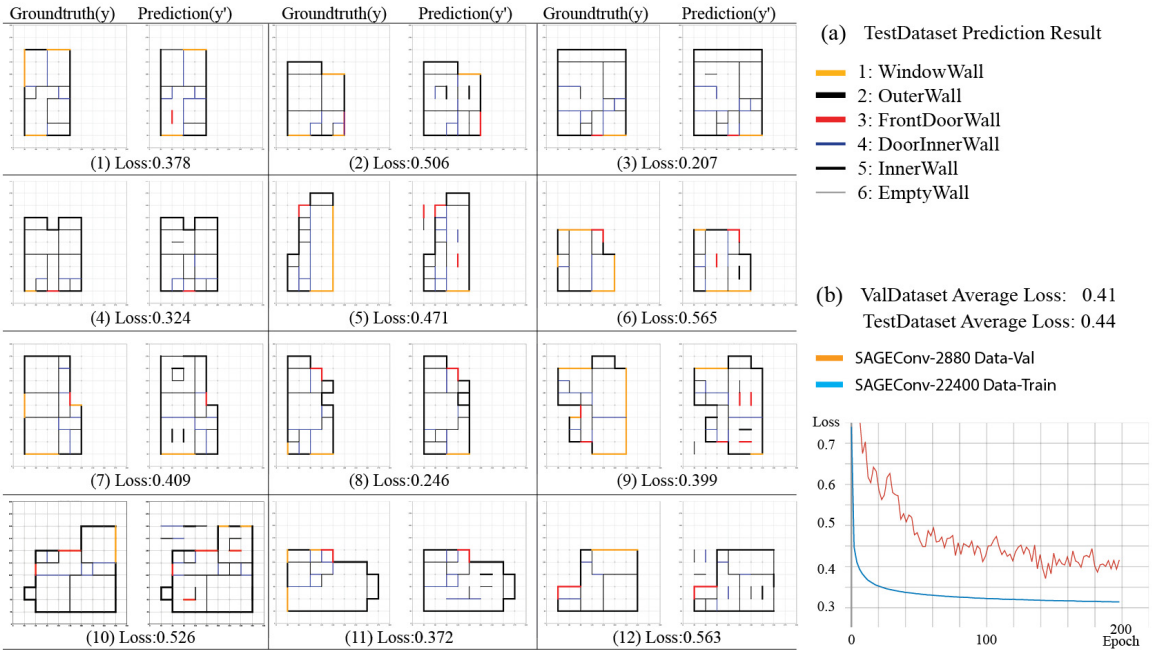

Fig. 7 Evaluation of Room GNN graph reconstruction result based on the test dataset

Fig. 8 demonstrates the generated room layouts of Room-Agent from various functional bubble diagrams and building boundary constraints. We analyzed the results using two metrics: Room Accessibility Index (RAI) and Room Enclosure Index (REI). In the generated results, the RAI is above 50% for all cases, indicating high accessibility. The first bubble diagram mapped to three building boundary layouts shows the highest room accessibility, with Figs. 8(1) reaching 100%. In terms of the REI, the spatial layouts show ideal room enclosure. For the third bubble diagram's generated results, Figs. 8(3) and (6) have an REI of 100%, indicating efficient room circulation. However, in Figs. 8(3) and (6), some wall types were not well predicted, such as multiple front doors in the entrance hall space. The second bubble diagram's overall generated results have lower RAI and REI indices. Figs. 8(5) and (8) have an RAI of 57%, with nearly half of the rooms lacking door connections. Fig. 8(8) has the lowest REI at 43%, where two living rooms lack outer boundaries and internal partitions, and the kitchen and storage room have become a single space without wall divisions. Overall, Room-Agent can effectively generate room layouts with detailed spatial divisions and high accessibility based on the provided functional bubble diagrams and building boundary constraints. However, there exists space for improvement in predicting wall types and handling layouts for spaces with special functions. These findings provide direction for further optimization of Room-Agent.

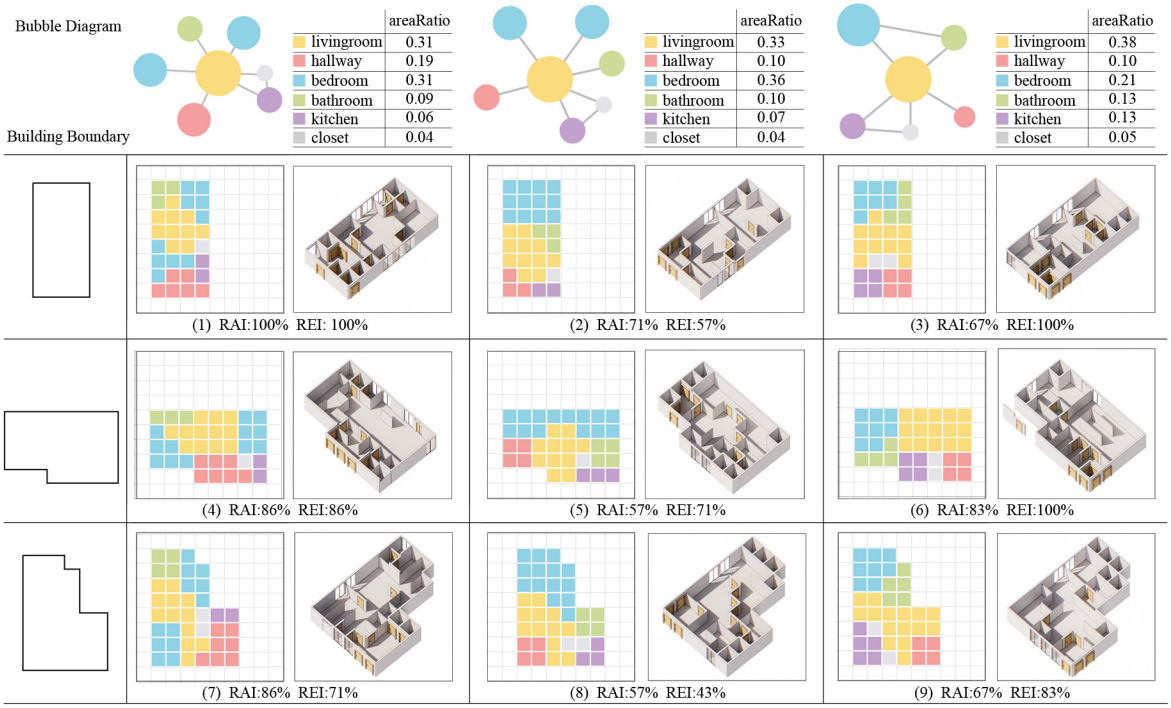

Fig. 8 Evaluation of Room-Agent generated result based on different bubble diagram and building boundary

Fig. 9 shows the experimental results of modifying the spatial size and layout relationships through textual input. In the first modification, Room-Agent successfully met the user's request to reduce the bathroom area and optimize the bedroom into a rectangular layout. The modified three-dimensional spatial layout exhibited a high room enclosure rate and accessibility. In the second modification, Room-Agent repositioned the kitchen as requested, with the closet occupying the rectangular space in the lower left corner. The shape of each room in the current design is almost square, except shapes of the bedroom and bathroom. The LLM within Room-Agent, functioning as a probabilistic model, generated new layout results causing the bedroom's shape to become irregular, meaning a lack of precise control. These results suggest that Room-Agent can comprehend textual requirements and translate them into spatial layout outcomes, while also introducing some additional variations in the layout results due to the probabilistic nature of the large language model.

Compared to existing AI models, Room-Agent can define specific rooms and wall partitions within those rooms based on the constraints of the building boundary (Fig. 10). In contrast, previous generative AI models relied solely on abstract functional bubble diagrams, making it difficult to control the shape of the generated spatial layouts accurately. Room-Agent possesses the ability to accurately control the spatial layout generation based on the bubble diagram and the building boundaries.

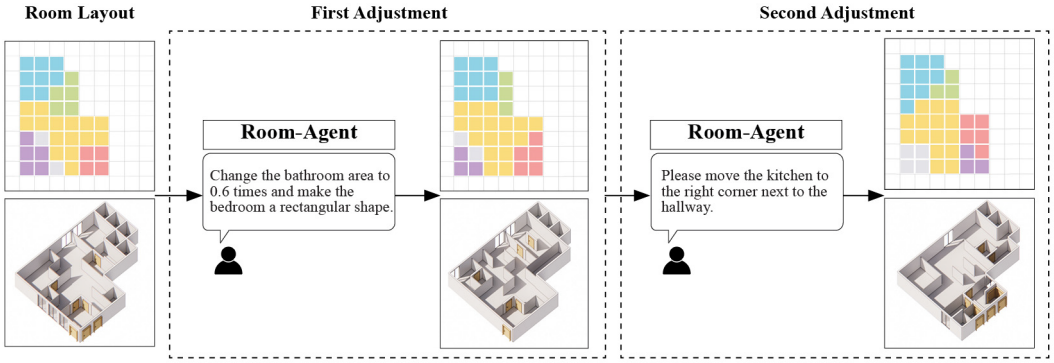

Fig. 9 Refining the room layout according user's requirement through natural language input

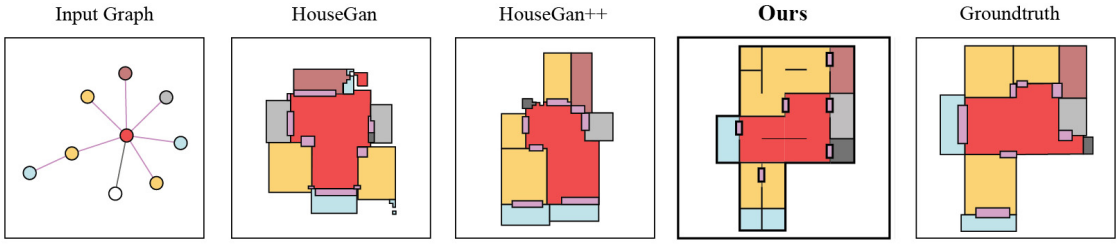

Fig. 10 Comparison of results between Room-Agent and existing AI models

5 Limitation and Future Work

Spatial layouts generated by the Room-Agent are constrained by the precision and diversity of the dataset. Experimental results indicate that some rooms fail to be with enclosed spaces, suggesting that the node features defining the outer boundaries need further refinement and diversification for training. Therefore, future improvements should focus on enhancing the accuracy and increasing the diversity of the dataset while providing more detailed features of each room. Additionally, developing adaptive adjustment methods to accommodate complex boundary constraints through advanced user feedback mechanism is another challenging task. Compared to traditional parameter control methods, using textual input to control architectural layouts demonstrates clear advantages in efficiency and simplicity. However, we also observe that there may be instability in the outputs of large language models when generating spatial layouts. This could lead the model to modify room layouts without fully understanding the user's needs, thus affecting the output. Introducing more functional components within LLM that are suitable for spatial layout methods will be an important area for further work. The rule-based functions and the trained GNN model embedded in Room-Agent have already shown their diverse possibilities. To further enhance its ability to handle complex tasks, future work could consider using Room-Agent as a backbone, connecting various AI models across modalities, including video and voice inputs. Thus, Room-Agent could not only manage the design of architectural spatial layouts but also provide users with a more comprehensive and personalized design service through the integration of multimodal data.

6 Conclusions

This study proposes a novel cross-modal 3D room layout generation method aimed at addressing the limitations of traditional and current AI-driven approaches. By integrating graph-based representations (such as bubble diagrams) and textual descriptions into an LLM-based agent framework, combined with GNN, our method can dynamically and flexibly generate 3D spatial layouts that meet architectural design requirements. The incorporation of text input not only enhances the model's ability to generate intelligent and semantically consistent layouts but also allows for real-time adjustments during the generation process as needed. Bounda-

ry constraints further ensure the practicality of the generated layouts in real-world applications. Experimental results show that our method can accurately reconstruct room layouts, particularly in scenarios with clear spatial boundaries and functional divisions. Although challenges remain when handling complex boundary conditions, the Room-Agent system demonstrates potential in adapting to a wide range of design requirements through iterative refinement based on user input. The successful application of cross-modal (text, graphics, and spatial data) integration highlights the potential of our approach to bridge the gap between AI-generated designs and conventional architectural practices, providing valuable insights for developing more efficient, flexible, and designer-friendly tools in the field of architectural design.

References

1. Aalaei M, Saadi M, Rahbar M, Ekhlassi A (2023) Architectural layout generation using a graph-constrained conditional Generative Adversarial Network (GAN). Automation in Construction 155:105053
2. Cao Y, Li S, Liu Y, Yan Z, Dai Y, Yu PS, Sun L (2023) A comprehensive survey of ai-generated content (AIGC): A history of generative AI from GAN to ChatGPT. arXiv preprint arXiv:230304226
3. Emmons P (1998) The cosmogony of bubble diagrams. In: 86th ACSA Annual Meeting and Technology Conference, Constructing Identity, Cleveland, Ohio, pp 420-425
4. Höllein L, Cao A, Owens A, Johnson J, Nießner M (2023) Text2Room: Extracting textured 3D meshes from 2D text-to-image models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 7909-7920
5. 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 (TOG) 39:118: 111-118: 114
6. Li Z, Zhang C, Wang X, Ren R, Xu Y, Ma R, Liu X (2024) 3DMIT: 3D multi-modal instruction tuning for scene understanding. arXiv preprint arXiv:240103201
7. Liang J, Zhong X, Koh I (2024) BRIDGING BIM AND AI: A Graph-BIM Encoding Approach For Detailed 3D Layout Generation Using Vari- ational Graph Autoencoder
8. Liang J, Zhong X, Koh I (2024) BRIDGING BIM AND AI: A Graph-BIM Encoding Approach For Detailed 3D Layout Generation Using Variational Graph Autoencoder. In: International Conference on Computer-Aided Architectural Design Research in Asia. Association for Computer-Aided Architectural Design Research in Asia, pp 221-230
9. 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. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 13632-13641
10. Siddiqui Y, Alliegro A, Artemov A, Tommasi T, Sirigatti D, Rosov V, Dai A, Nießner M (2024) MeshGPT: Generating triangle meshes with decoder-only transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 19615-19625
11. Tan W, Ding Z, Zhang W, Li B, Zhou B, Yue J, Xia H, Jiang J, Zheng L, Xu X (2024) Towards general computer control: A multimodal agent for red dead redemption ii as a case study. arXiv preprint arXiv:240303186
12. Vilesov A, Chari P, Kadambi A (2023) CG3D: compositional generation for text-to-3d via gaussian splatting. arXiv preprint arXiv:231117907
13. Zheng Z, Petzold F (2023) Neural-guided room layout generation with bubble diagram constraints. Automation in Construction 154:104962
14. 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. eCAADe, pp 431-440
15. Zhong X, Liang J, Li Y (2024) Building-Agent: A 3D generation agent framework integrating large language models and graph- based 3D generation model. In: International Conference on Education and Research in Computer Aided Architectural Design in Europe, Nicosia, Cyprus. eCAADe
16. Zhou X, Ran X, Xiong Y, He J, Lin Z, Wang Y, Sun D, Yang M-H (2024) GALA3D: Towards text-to-3D complex scene generation via layout-guided generative gaussian splatting. arXiv preprint arXiv:240207207